

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



Gestão Automática de Dadores, Transfusões e Banco de Sangue

Ana Raquel Ferreira Alqueidão

Mestrado em Engenharia Informática
Especialização em Sistemas de Informação

Versão Pública

Trabalho de projeto orientado por:
Prof. Doutor. João Diogo Ferreira
e pelo Doutor Paulo Paiva de Sousa

2019

Agradecimentos

Primeiramente, agradeço aos meus pais por fazerem de mim o que sou hoje. Obrigada por todo o apoio, motivação, persistência e amor incondicional ao longo deste percurso acadêmico, que me permitiram realizar os meus estudos nas melhores condições possíveis.

Ao meu namorado Carlos, por toda a paciência, incentivo, encorajamento e amor que foram fundamentais para elaboração deste relatório.

Ao meu irmão Ricardo, por todo o incentivo e persistência em procurar fazer sempre melhor durante todo este percurso acadêmico.

Aos meus orientadores, o Doutor Paulo Sousa e o Professor Doutor João Ferreira, por todo o apoio e conselhos dados que foram fundamentais para a elaboração deste relatório.

À Maxdata, pela oportunidade de integrar neste projeto na área da saúde, a todos os funcionários, em especial ao Paulo Sousa, Alexandre Correia e aos meus colegas de equipa de desenvolvimento, André Félix, Enoque Duarte, Rossana Naki, que foram fundamentais na minha integração na empresa.

A todos os meus amigos e colegas de curso, agradeço o companheirismo, amizade e positivismo ao longo desta longa caminhada, sem vocês não teria sido o mesmo.

Resumo

Em Portugal todos os dias são necessárias entre 900 a 950 unidades de sangue nos hospitais portugueses e há cerca de 209 mil dadores de sangue. Face a estes números é fundamental que haja uma boa gestão e rastreio das unidades sanguíneas provenientes de dadores e do seu processamento para serem transfundidas de maneira segura a pacientes que as necessitem.

Neste relatório encontra-se descrito o trabalho realizado durante o estágio feito na empresa Maxdata, onde fui integrada na implementação do *software Clinidata® Transfusion*, que vai ser integrado no *software Clinidata®*. É um sistema *web* de gestão de informação de dadores, transfusões e banco de sangue, dirigido aos enfermeiros, médicos e técnicos de laboratório. Implementado para atender as necessidades de negócio e fluxo de trabalho do processo transfusional, nomeadamente a gestão de doações, de transfusões e de unidades sanguíneas (banco de sangue) e permite integrações com serviços externos, permitindo a comunicação entre serviços.

A implementação deste módulo teve como base um conjunto de requisitos recolhidos e documentados por analistas seniores da Maxdata e seguiu um modelo de desenvolvimento ágil de modo a responder rapidamente a mudanças de requisitos. O módulo foi implementado segundo o padrão arquitetural *Model-View-Presenter*, organizado em três camadas (dados, negócio e apresentação) e com auxílio de várias *frameworks* (*Google Web Toolkit*, *Spring* e *Hibernate*).

Palavras-chave: Gestão de dadores, transfusões e banco de sangue, *Clinidata® Transfusion*, Web, Software, *Model-view-presenter*

Abstract

In Portugal, between 900 and 950 blood units are needed every day in Portuguese hospitals and there are about 209,000 blood donors. Given these numbers, it is essential that the donor blood units are well managed and screened and processed to be safely transfused to patients in need.

This report describes the work done during the internship at Maxdata, where I was integrated in the implementation of *Clinidata*® Transfusion software, which will be integrated into *Clinidata*® software. It is a web-based donation, transfusion and blood bank information management system, to be used by nurses, doctors and laboratory technicians. Implemented to meet the business needs and workflow of the transfusion process, including donation, transfusion and blood unit (blood bank) management and allows integrations with external services, allowing communication between services.

The implementation of this module was based on a set of requirements collected and documented by senior *Maxdata* analysts and followed an agile development model to respond quickly to changing requirements. The module was implemented by the Model-View-Presenter architectural standard, organized into three layers (data, business and presentation) was used and with the aid of several *frameworks* (*Google Web Toolkit*, *Spring* and *Hibernate*).

Keywords: web-based donation, transfusion and blood bank information management, *Clinidata*®*Transfusion* System, Web, Software, Model-View-Presenter

Índice

Capítulo 1	Introdução	21
1.1	Motivação.....	21
1.2	Objetivos.....	22
1.3	Instituição de acolhimento.....	23
1.3.1	Os seus produtos	23
1.4	Organização do documento	25
Capítulo 2	Enquadramento.....	27
2.1	Contexto do trabalho	27
2.1.1	Processo transfusional.....	27
2.1.1.1	Componentes sanguíneos	28
2.1.1.2	Doação.....	29
2.1.1.3	Transfusão	32
2.1.2	Contexto de um software na área da saúde	34
2.2	Gestão de Projeto.....	34
2.2.1	As Pessoas.....	35
2.2.2	O Produto	36
2.2.3	O Processo.....	36
2.2.4	O Projeto	38
2.3	Trabalho relacionado	42
Capítulo 3	Análise de requisitos	47
3.1	Requisitos	47
3.1.1	Requisitos Funcionais	48
3.1.2	Requisitos Não funcionais.....	50
3.2	Modelo de Dados.....	51
3.2.1	Entidades e Relações.....	52
3.3	Diagrama de casos de uso e de fluxo.....	56
Capítulo 4	Implementação	58
4.1	Ferramentas utilizadas	58
4.2	Fluxo de sequência	63
4.3	Arquitetura do Sistema	64
4.3.1	Camadas de dados	66
4.3.2	Camada de negócio	68
4.3.3	Camada Cliente	71

4.3.4	<i>Packages</i> criados no Clinidata® Transfusion	72
4.4	Casos de uso	73
4.4.1	Configuração de ecrã.....	73
4.4.2	Gestão de doações	76
4.4.3	Detalhe de doação	78
4.4.4	Decomposição de unidades	78
4.4.5	Medições vitais.....	80
4.4.6	Validação de unidades.....	82
4.4.7	Gestão de transfusões	84
4.4.8	Detalhe de transfusão	85
Capítulo 5	Considerações finais.....	87
5.1	Trabalho realizado	87
5.2	Dificuldades encontradas.....	88
5.3	Trabalho futuro.....	88
Capítulo 6	Bibliografia	89
Capítulo 7	Anexos	92
7.1	Diagrama de relações de dadores	92

Acrónimos

AJAX Asynchronous JavaScript and XML

BD Base de Dados

CRUD Create Read Update and Delete

DAO Data Access Object

FCUL Faculdade de Ciências da Universidade de Lisboa

FK's Foreign Keys

GWT Google Web Toolkit

HIS Hospital Information System

HL7 Health Level 7

HQL Hibernate Query Language

HTML Hypertext Mark-up Language

I&D Investigação e Desenvolvimento

IPST Instituto Português do Sangue e da Transplantação

LIS Laboratory Information System

MVP Model-View-Presenter

ORM Object Relational Mapping

PEI Projeto de Engenharia Informática

POJO Plain Old Java Object

RPC Remote Procedure Call

SGBD Sistema de Gestão de Base de Dados

SGBDR Sistema de Gestão de Base de Dados Relacional

SNS Serviço Nacional de Saúde

SOA Service Oriented Architecture

SOAP Simple Object Access Protocol

SOFEA Service Oriented Front-End Architecture

SQL Structured Query Language

ST Sangue Total

SW Software

UML Unified Modelling Language

XML Extensible Mark-up Language

Lista de figuras

- Figura 1** - Produtos disponibilizados pelo novo software *Clinidata*®
- Figura 2** - Etapas do processo transfusional
- Figura 3** - Compatibilidade Ab0-Rh
- Figura 4** - Etapas de uma doação (Elaboração própria)
- Figura 5** - Etapas de uma transfusão de sangue
- Figura 7** - Iterações no modelo de desenvolvimento ágil usado na Maxdata
- Figura 8** - Mapa de *Gantt*
- Figura 9** - Exemplo de um ecrã *Sunquest Blood Bank*™
- Figura 10** - Modelo de domínio
- Figura 11** - Diagrama de casos de uso do módulo de dadores e da gestão de transfusões (Elaboração própria)
- Figura 12** - Exemplo de uma *HQL Query* realizada neste PEI
- Figura 13** - Exemplo de uso das anotações de Spring para injeções de dependências deste PEI
- Figura 14** - Exemplo de um *changeSet* criado no ficheiro “dbChangeLog”
- Figura 15** - Diagrama de fluxo do módulo de dadores
- Figura 16** - Diagrama de interacções entre as camadas aplicacionais
- Figura 17** - Vista geral dos packages do Clinidata
- Figura 18** - *Changeset* para a criação de uma tabela e a sua chave primária
- Figura 19** - Modelo de domínio criado para a tabela de doação “DBB_DONATION”
- Figura 20** - Versão geral do package “shared/bloodBank”
- Figura 21** - Acrescentado o novo “donationService” ao ficheiro *rpc-services-servelet.xml*
- Figura 22** - Classes “IDonationBC” e “IDonationBCAsync”
- Figura 23** - Classe “DonationBC”
- Figura 24** - Classe “DonationDAO”

Figura 25 - Exemplo de hqlQuery criado no *repository* “DonationDAO”

Figura 26 - Adicionar o novo “donationService” ao ficheiro *rpc-services-servelet.xml*

Figura 27 - Classes “IDonationBC” e “IDonationBCAsync”

Figura 28 - Classe do “DonationBC”(*service* de doações)

Figura 29 - Exemplo de método criado no “DonationBC” (*service* das doações)

Figura 30 - Interface “IDonationDAO”(*interface do repository* das doações)

Figura 31 - Classe “DonationDAO” (*repository* das doações)

Figura 32 - Visão geral do *package* “server”

Figura 33 - Visão geral do *package* “shared”

Figura 34 - Padrão arquitetural MVC vs. MVP

Figura 35 - Visão geral do *package* “cliente” do banco de sangue

Figura 36 - Visão geral do *package* “cliente” do banco de sangue

Figura 37 - Ecrã de “login” da aplicação *Clinidata®*

Figura 38 - Ambiente de trabalho do utilizador na aplicação *Clinidata®* (esquerda) e exemplo da pesquisa de um ecrã gerado (direita)

Figura 39 - Exemplo de configuração da geradora do banco de sangue para exibir o ecrã de “Gestão de doações”

Figura 40 - Menu inicial do software *Clinidata®*

Figura 41 - Exibição do ecrã “Gestão de doações” com id 39655

Figura 42 - Ecrã “Pesquisa” (cima direita) e *grid* de resultados (baixo) da Gestão de doações

Figura 43 - Ecrã aberto depois de ter clicado em “Imprimir” na fig.42 (esquerda) e *pop-up* de impressão após ter clicado em “Imprimir” (direita)

Figura 44 - Ecrã “Detalhe de doação”

Figura 45 - Ecrã “Decomposição de unidades”

Figura 46 - Resultado após ter criado a unidade de componente de Sangue total

Figura 47 – Ecrã “Inserir unidade” (esquerda) e Ecrã “Editar unidades” (direita)

Figura 48 - Ecrã de impressão de etiquetas “iniciais” das unidades

Figura 49 - Etiqueta para a unidade segundo a norma IBST128

Figura 50 - Ecrã de “Medições vitais” (âmbito doações)

Figura 51 - Ecrã de “Medições vitais” quando é clicado em “alterar” com uma medição selecionada vital (âmbito doações)

Figura 52 - Ecrã de “Medições vitais” (âmbito transfusões)

Figura 53 - Ecrã de “Validação de unidades”

Figura 54 - Ecrã “Alterar unidade”

Figura 55 - Ecrã de impressão de etiquetas “finais” da unidade

Figura 56 - Ecrã de “Gestão de transfusões” (Pesquisa de transfusões)

Figura 57 - Ecrã de “Gestão de transfusões” (*Grid* de resultados)

Figura 58 - Ecrã de “Detalhe de transfusão”

Lista de tabelas

Tabela 1 - Tipos de componentes sanguíneos

Tabela 2 - Comparação entre as diferentes aplicações transfusionais

Tabela 3 - Requisitos funcionais associados ao módulo dadores do *Clinidata®Transfusion*

Capítulo 1 Introdução

O trabalho descrito neste relatório foi realizado no âmbito do Projeto de Engenharia informática (PEI), integrado no Mestrado em Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa (FCUL). O projeto descrito foi desenvolvido durante o estágio na empresa Maxdata. Este capítulo apresenta uma contextualização do projeto, as suas motivações, objetivos, a instituição de acolhimento e as diferentes organizações do documento.

1.1 Motivação

Em Portugal todos os dias são necessárias entre 900 a 950 unidades de sangue nos hospitais portugueses e há cerca de 209 mil dadores de sangue [1]. Face a estes números é fundamental que haja na área da saúde sistemas de informação que permitam a gestão e rastreio das unidades sanguíneas provenientes de dadores, do seu processamento para serem transfundidas de maneira segura a pacientes que as necessitem [2]. Uma das motivações é, portanto, a disponibilização de um sistema de *sw* (*software*) de gestão de informação, *Clinidata® Transfusion*, que vai de acordo com os requisitos necessários para os profissionais de saúde executarem com maior facilidade as suas funções em todo o processo transfusional.

Outra motivação prende-se com a constante evolução tecnológica, os surgimentos de novos paradigmas de desenvolvimento e realidades, mudanças de *hardwares*, sistemas operativos, ferramentas utilizadas, alterações aos requisitos, levam a que um *sw* se tenha que adaptar e evoluir ao longo de diferentes gerações, sendo substituído pela implementação de um novo que corresponda aos novos paradigmas [3]. Neste sentido, existem cada vez mais empresas a fazer o *shift* para *cloud*, como é o caso da Maxdata, que tem vindo a transferir as suas aplicações desenvolvidas anteriormente em *Visual Basic 6* e *VB.net* para o mundo da *Web* (seção 1.3.1), adaptando-se aos novos paradigmas de desenvolvimento, levando a uma reanálise e reprogramação dos componentes já desenvolvidos, de modo a tornar o seu software preparado para ser executado em *cloud*, independente das plataformas usadas pelos clientes e adaptável aos requisitos e preferências dos clientes. No contexto de aplicações na área da saúde, podemos definir a computação em *cloud*, como uma infraestrutura orientada a aplicações e baseada em serviços, onde os recursos de *health-IT*² são agrupados,

² *health-IT*(*Health Information technology*): <https://www.healthit.gov/faq/what-health-it>

permitindo que os serviços sejam amplamente implementados ou rapidamente compartilhados em resposta a mudanças de requisitos de negócio. Do ponto de vista do utilizador, os recursos internos e os processos de aplicação são-lhes transparentes, permitindo que se foquem nos registos de saúde padronizados e nos recursos familiares da tecnologia de informação em saúde [4]. A computação em *cloud* é um paradigma de computação em que empresas e utilizadores conseguem aceder às aplicações em todo o mundo por meio de qualquer navegador *web*, um sistema virtualizado que disponibiliza plataformas e sw's como serviços [5].

No contexto deste PEI surge a implementação do *Clinidata® Transfusion*, que representa a evolução de um produto anterior da Maxdata para os paradigmas atuais de *cloud*, para ser integrado no *Clinidata®*.

1.2 Objetivos

O *Clinidata® Transfusion* é um sistema *Web* de gestão de dadores, transfusões e banco de sangue, implementado segundo as boas práticas e normas do processo transfusional [6], oferecendo uma solução completa para os serviços hospitalares: gestão de doações, abrange todo o processo desde a consulta do dador até à validação das unidades finais para ficarem disponíveis para a transfusão, gestão de transfusões, abrange todo o processo desde o pedido de transfusão pelo médico no serviço, até à administração da unidade, incluindo o registo e investigação de possíveis reações adversas e a gestão de unidades que contempla a troca de unidades sanguíneas com entidades externas (envio e receção).

O principal objetivo deste PEI assentou na implementação do módulo de gestão de doações, para integrar no *Clinidata® Transfusion*. Fui responsável pela criação dos modelos de domínio inerentes às doações, implementação do serviço que representa as doações e implementação dos ecrãs “Gestão de doações”, “Detalhe de doação”, “Decomposição de unidades”, “Validação de unidades,” “Gestão de Transfusões”, “Detalhe de transfusão” e “Medições vitais”, seguindo um modelo *MVP*.

Os principais objetivos do módulo de gestão de doações são:

- Identificar com antecedência doadores não aptos
- Reduzir falhas no processo de colheita
- Integrar as etapas de identificação, pré-triagem e doação (colheita)
- Reduzir o tempo de espera e número de formulário manuais
- Registo das medições vitais
- Registo da decomposição de unidades
- Registo da validação de unidades
- Impressão de etiquetas para as unidades sanguíneas doadas

- As criações dos dadores, da doação, das unidades produzidas e os seus estados, são atualizadas automaticamente através do ecrã de consulta, colheita, validação, decomposição e medições vitais

1.3 Instituição de acolhimento

A instituição de acolhimento para o desenvolvimento deste PEI é a Maxdata⁴ software, S.A. uma empresa que concebe, desenvolve, instala e mantém software para a área de saúde há mais de 40 anos. A Maxdata comercializava aplicações independentes para várias áreas do hospital. Nos últimos anos quis adaptar os seus produtos aos novos paradigmas, como o *cloud computing* [7] e funcionalidades emergentes na nova era de Web 2.0 [8], levando a uma reengenharia dos seus produtos de modo a desenvolver uma única aplicação *100%web*, *cross-plataform*, *cross-browser* e *cloud-ready*, o software *Clinidata®*. No âmbito deste PEI, o projeto foi realizado na sede localizada no Carregado. Foi integrada na implementação do módulo de *sw Clinidata®Transfusion* que permite a gestão automática de dadores, transfusões e banco de sangue.

1.3.1 Os seus produtos

A Maxdata comercializava aplicações independentes para várias áreas hospitalares, atualmente, aposta na internacionalização pelo que investiu dezenas de milhar de horas no reengenharia dos seus produtos de modo a desenvolver uma única aplicação *cross-plataform*, *cross-browser* e *cloud-ready* que integra todas as aplicações em ambiente *web*, adaptando os seus produtos a novos paradigmas de computação, como o *cloud computing* [7] e funcionalidades emergentes na nova era de Web 2.0 [8]. Na conceção, codificação e teste do novo *sw Clinidata®*, um *sw 100% web*, que permite a utilização de sistemas operativos e bases de dados *open source*, inclui a migração automática de aplicações concorrentes e está preparado para ser utilizado a partir de *cloud*. Apesar de o meu trabalho se incidir sobre o software *Clinidata® Transfusion*, este está integrado com outros módulos do software *Clinidata®* que interagem entre si, seja de um ponto de vista funcional ou de interface, por isso o módulo foi desenvolvido em conformidade com os restantes módulos, para que a Maxdata mantenha uma interface coesa para contribuir para o aumento da sua usabilidade e experiência de utilizador.

⁴ Maxdata: <http://www.maxdata.pt>



Figura 1. Produtos disponibilizados pelo novo software *Clinidata®* (consultado em <http://www.maxdata.pt/site/uploads/ckeditor/datanews61.pdf>)

A nova aplicação (*Clinidata®*) é composta por 3 módulos tal como mostrado na figura 1:

- **CLINIdATA® LIS** (*Laboratory Information System*): corresponde à adaptação dos antigos *Clinidata®XXI* (patologia clínica, é um sistema de gestão de laboratórios de análises clínicas e de diagnóstico, nos aspetos técnicos e financeiros), *Clinidata®ANP* (anatomia patológica, é um sistema de gestão global do laboratório de anatomia patológica, incluindo requisição electrónica de exames, e a consulta dos relatórios nos serviços hospitalares, permitindo assim uma interação online entre o laboratório de anatomia patológica e os serviços clínicos) e *Clinidata®NET* (requisição electrónica, é um sistema *Web* que usa a Intranet do hospital, clínica ou laboratório para ligar os médicos e os serviços aos laboratórios de diagnóstico);
- **CLINIdATA® Vigilant** (vigilância epidemiológica): ferramenta de vigilância epidemiológica vocacionada para o controlo de infeções e resistência a antimicrobianos. Este módulo nasceu com o novo *Clinidata®*, permite a identificação, deteção, prevenção e controlo de doenças, a nível pessoal e coletivo, abrangendo toda a instituição médica;
- **CLINIdATA® Transfusion** (gestão do processo transfusional): módulo na qual se encontra inserido neste PEI, corresponde à adaptação do produto *Clinidata®BST* [10] e passou a fornecer serviço de gestão de doações. Trata-se de um sistema *web* de gestão automática de dadores, transfusões e banco de sangue, dirigido ao laboratório/médico/enfermeiro que oferece uma solução que contempla todas as etapas inerentes ao processo transfusional, garantindo

segurança, fiabilidade e rastreabilidade. O módulo desenvolvido vai ser integrado no sw *Clinidata*® e teve maior incidência no módulo de gestão de dadores.

1.4 Organização do documento

Este documento está organizado do seguinte modo:

- **Capítulo 1** – introdução do PEI desenvolvido, as suas motivações, objetivos e instituição de acolhimento;
- **Capítulo 2** - contextualização do projeto, introduzindo os conceitos e processos necessários, é apresentado a gestão do projeto e o trabalho relacionado;
- **Capítulo 3** - identifica e descreve os requisitos do projeto, o seu modelo de dados e a arquitetura da aplicação;
- **Capítulo 4** – descreve o processo de implementação do projeto, incluindo as diferentes ferramentas utilizadas, a arquitetura em camadas e casos de usos dos ecrãs implementados no âmbito deste projeto, para integrar no *Clinidata*® *Transfusion*;
- **Capítulo 5** – são apresentadas as conclusões sobre o trabalho desenvolvido, as dificuldades encontradas e o trabalho futuro.

Capítulo 2 Enquadramento

2.1 Contexto do trabalho

O *Clinidata@Transfusion* é um sistema *web* de informação de gestão de dadores, transfusões e banco de sangue. Como tal, encontra-se detalhado nesta secção as diferentes etapas inerentes ao processo transfusional e é apresentado o contexto de uma aplicação de sistemas de gestão de informação nesta área.

2.1.1 Processo transfusional

O processo transfusional é composto por **componentes sanguíneos**, obtidos através da dádiva de sangue dos **dadores**, que são submetidos a um conjunto de exames, decompostos e quando validados, tornam-se disponíveis para serem usados numa **transfusão** a um ou mais pacientes. As diferentes etapas de um processo transfusional encontra-se na figura 2.

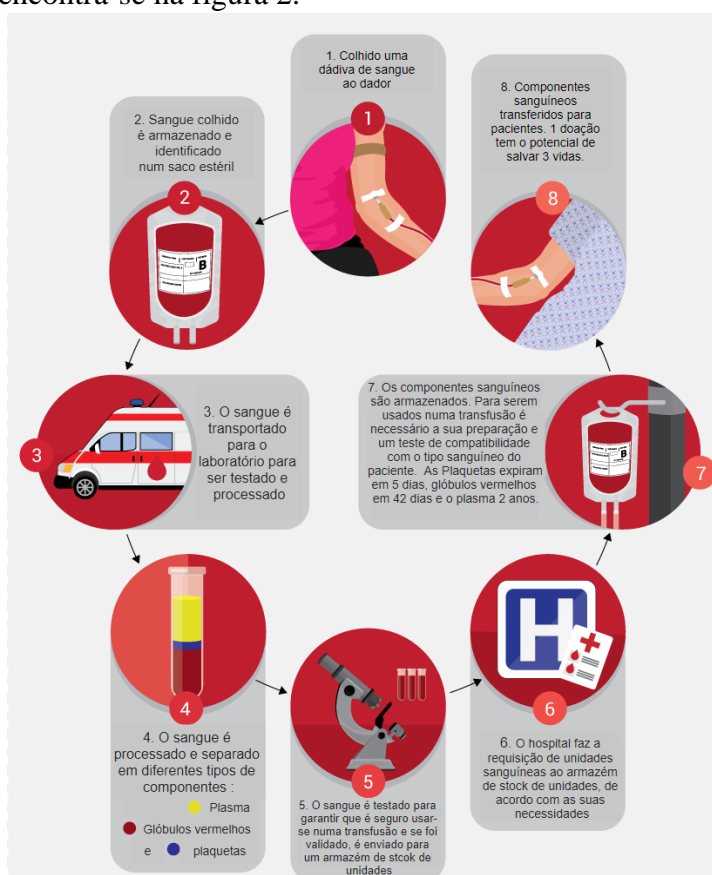


Figura 2. Etapas do processo transfusional (Elaboração própria)

2.1.1.1. Componentes sanguíneos

O sangue possui três tipos fundamentais de células: os **glóbulos vermelhos**, **plaquetas** e **glóbulos brancos**. Os glóbulos vermelhos contêm hemoglobina, responsável por transportar o oxigênio a partir dos pulmões para todo o corpo e, inversamente, o dióxido de carbono a partir dos tecidos para ser eliminado pelos pulmões. As plaquetas são essenciais para a coagulação sanguínea e os glóbulos brancos atuam como mecanismo de defesa contra infecções. O sangue também é constituído por plasma (parte líquida) onde existem proteínas que ajudam na coagulação do sangue (fatores da coagulação), entre outras funções.

Numa doação, é realizada a colheita de uma dádiva que vem sobre a forma de componente de sangue total (ST), que pode dar origem a hemocomponentes e a hemoderivados. Os hemocomponentes são os produtos gerados em serviços de hemoterapia através de técnicas de centrifugação que permitem o fracionamento da bolsa de sangue total em concentrado de glóbulos vermelhos, concentrado de plaquetas, plasma fresco congelado e crioprecipitado. Os hemoderivados são produzidos em escala industrial através do processamento do plasma, que é submetido a um fracionamento que permite extrair proteínas específicas. [11]. As diferentes componentes sanguíneas que se podem obter a partir de uma dádiva estão apresentadas na Tabela 1.

Abreviatura	Tipo de Componente
ST	Sangue Total
CP	Concentrado de plaquetas
PFC	Plasma fresco congelado
CE	Concentrado de glóbulos vermelhos
CRIO	Crioprecipitado
CUP	Concentrado Unitário Plaquetas
CPP	Pool Plaquetas Desleucocitadas
PFI	Plasma Fresco Inativado
CLP	Complexo Protrombínico Plasma
CLPP	Complexo Protrombínico Plasma
CEA	Concentrado de glóbulos vermelhos A

Tabela 1. Tipos de componentes sanguíneos

Os glóbulos vermelhos contêm antígenos, estes podem ser do tipo A e/ou B, dando origem aos nomes A, B, AB (contém os dois tipos) e 0 (não contém nenhum tipo). O sistema Rh segue a mesma lógica do sistema AB0, Rh é um antígeno designado de Antígeno D, que pode ou não estar presente nas membranas dos glóbulos vermelhos, se estiver presente então o paciente é classificado como Rh positivo, caso contrário é classificado como Rh negativo. [12]. Após a decomposição da unidade colhida nos diferentes componentes, são realizados exames às unidades produzidas, que se forem validadas, prosseguem para um armazém de stock de unidades. Quando um serviço necessita de unidades sanguíneas, faz uma requisição ao armazém de *stock*.

Uma componente antes de poder ser utilizada numa transfusão, é realizado um teste de *crossmatch* onde se realiza um teste de compatibilidade entre a unidade a ser transfundida e o grupo sanguíneo do paciente que a vai receber, sendo obrigatório que haja uma compatibilidade AB0/Rh que pode ser consultada na figura 3.

		DADOR							
		O-	O+	B-	B+	A-	A+	AB-	AB+
PACIENTE	AB+	✓	✓	✓	✓	✓	✓	✓	✓
	AB-	✓		✓		✓		✓	
	A+	✓	✓			✓	✓		
	A-	✓				✓			
	B+	✓	✓	✓	✓				
	B-	✓		✓					
	O+	✓	✓						
	O-	✓							

Figura 3. Compatibilidade Ab0/Rh (adaptado de ⁵)

2.1.1.2 Doação

Em Portugal, os dadores apresentam-se por iniciativa própria a um local de colheita de sangue, podendo recorrer ao site de dador, que identifica quando e onde poderá doar

⁵ <http://www.oswaldocruz.com/site/noticias-em-geral/noticias/saiba-mais-sobre-karl-landsteiner-o-medico-que-descobriu-os-tipos-sanguineos-e-o-fator-rh>

sangue, com base na sua localização, e lhe disponibiliza a funcionalidade de definição da rota para o local da dádiva selecionado; ou pode deslocar-se presencialmente a um local de colheita de sangue fazendo-se acompanhar de um documento de identificação válido (Bilhete de identidade/cartão de cidadão). [13].

Uma doação de sangue é o processo pela qual um doador voluntário disponibiliza uma dádiva (porção do seu sangue para ser colhido) para uso subsequente numa transfusão sanguínea. O processo de doação de sangue abrange várias etapas sendo elas: consulta de doação, colheita, decomposição, validação e disponibilização das unidades sanguíneas.

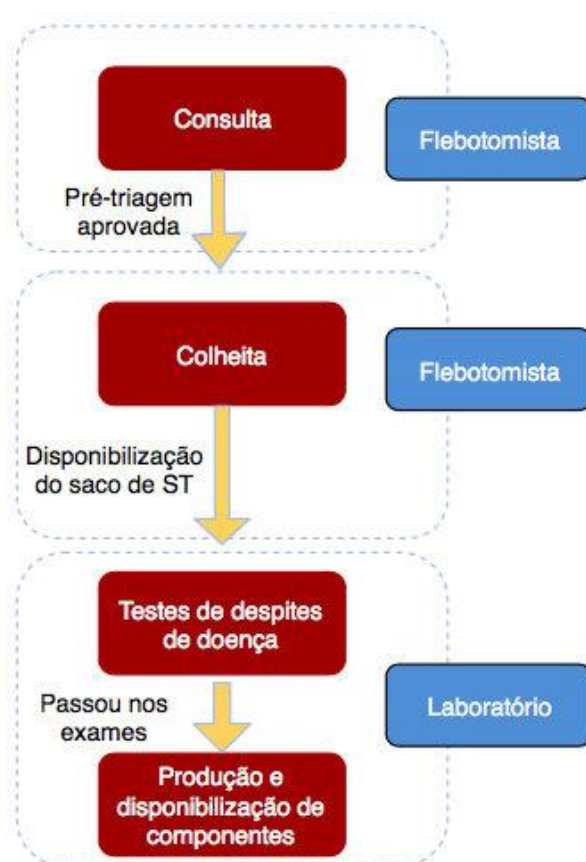


Figura 4. Etapas de uma doação (Elaboração própria)

Como é possível observar na Figura 4 uma doação requer:

1. **Consulta de doação:** os dadores apresentam-se por iniciativa própria a um local de colheita de sangue e seguem para uma consulta com uma flebotomista, entidade responsável pela recolha de amostras de sangue de pacientes para análises, transfusões ou outros tipos de investigação. É realizado uma **pré-triagem** através de um questionário ao dador, sobre o seu histórico médico, hábitos e condições de saúde. É definido o tipo de doação, podendo ser: autóloga quando se trata de

unidades de sangue que só podem ser administradas ao próprio dador, homóloga quando podem ser administradas a qualquer paciente, ou sangria quando o sangue extraído deve ser imediatamente destruído, uma vez que a sua finalidade é retirar algumas substâncias do organismo ao dador. Durante a pré-triagem, é necessário realizar medições vitais ao dador, medindo-se a pressão arterial, a pulsação, temperatura e é feito um teste de anemia. O teste de anemia mais conhecido como “picadela” surge para garantir a saúde do dador, é com essa pequena gota de sangue que determinam a hemoglobina do dador. Com a “picadela” os médicos conseguem saber se o dador tem anemia ou se fica com anemia depois de doar 450ml de sangue. Caso o dador esteja dentro dos requisitos exigidos, tais como ter um bom estado de saúde, hábitos saudáveis de vida, peso superior ou igual a 50kg e idade compreendida entre os 18 e 65 anos, entre outros requisitos, [2], a flebotomista aprova o dador, prosseguindo para a etapa seguinte.

2. **Colheita:** realizada pela flebotomista, corresponde à etapa principal do processo de doação, consiste em retirar ao dador cerca de 400ml a 450ml de amostra do sangue, que vem sobre a forma de componente sangue total (ST). Atualmente em Portugal, o sangue é colhido para um sistema de quatro sacos. Apenas um, o saco mãe, recebe a dádiva de sangue. Em seguida, o saco é enviado para o laboratório para seguir para a etapa seguinte.
3. **Teste de despistes de doenças:** realizado no laboratório pelos técnicos de laboratório, é retirado uma pequena amostra do sangue doado para execução de testes laboratoriais para doenças infecciosas transmissíveis por transfusão e é realizado um teste para determinar o tipo sanguíneo, que corresponde à determinação do grupo AB0-Rh do dador [14]. Depois de colhido, determinado o seu grupo sanguíneo e testado relativamente a diversas doenças, se não foi detetado nenhum problema, a unidade é validada e prossegue para a etapa seguinte.
4. **Produção e disponibilização de componentes:** realizado no laboratório pelos técnicos de laboratório, onde é feita a decomposição da unidade de componente de ST doada, ou seja, é realizada uma divisão em vários componentes que podem ser transfundidos, processados, embalados e armazenados em condições próprias para a sua conservação. De seguida são efetuados exames específicos às unidades decompostas, sendo necessário uma validação das unidades do técnico de laboratório. Caso tenha passado nos exames específicos, a unidade é validada tornando-se disponível para ser usada numa transfusão, caso contrário a unidade é destruída.

2.1.1.3 Transfusão

Uma transfusão de sangue ou de componentes sanguíneos consiste na administração a um paciente, através de um cateter venoso, de componentes do sangue (glóbulos vermelhos, plaquetas, plasma) ou de substâncias que intervêm na formação e funções normais do sangue (fibrinogénio, fatores de coagulação, entre outros), provenientes de outra pessoa (o dador).

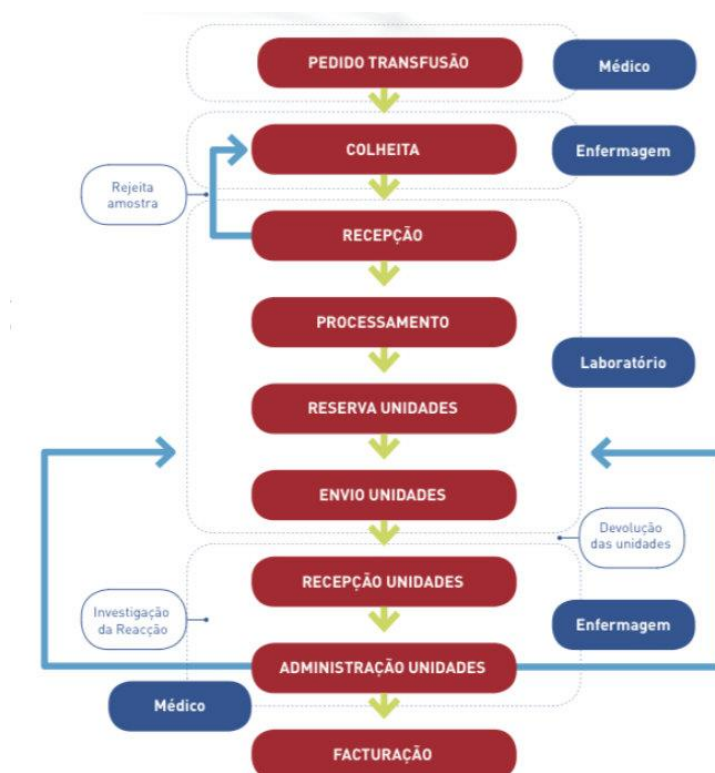


Figura 5. Etapas de uma transfusão de sangue (consultado em http://www.maxdata.pt/site/uploads/produtos/documentos/B9EEF5B1-FB010_1.pdf)

As etapas do processo transfusional, tal como mostrado na figura 5, passam por:

1. **Pedido de transfusão**, realizada pelo médico de serviço, nos módulos de urgência, internamento ou consulta. Este é responsável por especificar o tipo de urgência da transfusão que vai ter influência na determinação automática da data prevista de administração e realizar a requisição dos componentes necessários;
2. **Colheita**, para cada transfusão é necessário fazer uma colheita de sangue ao paciente para determinar o seu grupo sanguíneo. Em Portugal, quando um doente aparece pela primeira vez, colhem-se sempre duas amostras para se fazer o teste ao Grupo AB0 Rh e fazem-se duas análises se os dois resultados coincidirem

considera-se esse resultado. Após a colheita, as amostras são enviadas para o laboratório;

3. **Receção da amostra**, é realizada no laboratório, onde são recebidas as dádivas.
4. **Processamento das unidades**, é realizada no laboratório, o processamento é realizado por unidade de componente e consiste em selecionar as unidades a fornecer com base no pedido de transfusão, da disponibilidade do banco de sangue e da compatibilidade das unidades com o paciente. Após as análises realizadas para determinar o grupo AB0 Rh do paciente, o médico pesquisa pelas unidades de componentes em *stock* que sejam compatíveis com o grupo sanguíneo do paciente. É possível fracionar uma componente, ou seja, dividir a unidade em duas ou mais frações.
5. **Reserva de unidades**, é realizada após ter sido determinado o grupo AB0 Rh do paciente, onde o médico é responsável por selecionar uma unidade do componente do stock de unidades compatível com a componente requisitada no pedido de transfusão.
6. **Envio de unidades**, é realizada pelo enfermeiro que se desloca ao laboratório para ir buscar as unidades preparadas. O técnico regista no *Clinidata* as unidades enviadas e para que local é para ser enviada e é enviada uma notificação informando que as unidades se encontram em transporte.
7. **Receção de unidades**, é realizada pelo médico do serviço, que tem disponível uma lista de unidades já disponíveis para administração, e envia um enfermeiro para ir ao laboratório buscá-las. O enfermeiro regista a receção das respetivas unidades no serviço.
8. **Administração de unidades**, é necessário identificar a hora em que se inicia e termina a transfusão. Este ato é realizado pela equipa de enfermagem, e antes da administração da unidade deve haver a confirmação por dois enfermeiros diferentes. A transfusão pode corresponder à administração de várias unidades, assim o início da transfusão corresponde ao início da administração da primeira unidade e o final à administração da última unidade. Por unidade, são realizadas medições dos sinais vitais, obrigatoriamente uma ao início e outra no final, com 15 minutos de intervalo. Na administração de unidades podem ocorrer reações adversas, nesse caso, é lançado um alarme para o laboratório. Na sequência de uma reação adversa, é realizada uma investigação pelo médico que corresponde a um pedido de um conjunto de análises.

2.1.2

Contexto de um software na área da saúde

Um sistema de informação de gestão de banco de sangue, dadores e transfusões necessita de estar integrado com outros sistemas informáticos, nomeadamente:

- **HIS** (*Hospital Information System*): para obtenção/envio de dados sobre os pacientes alvo das transfusões;
- **LIS** (*Laboratory Information System*): para pedido/receção de análises ao sangue dos dadores e dos pacientes alvo das transfusões.
- **Autoridades reguladoras** das dádivas e transfusões de sangue: para envio de alertas sobre situações anómalas (por ex., dadores com doenças contagiosas) e para envio de estatísticas sobre unidades de sangue consumidas/disponíveis. Em Portugal, esta autoridade é a IPST (Instituto Português do Sangue e da Transplantação). Responsável por garantir e regular, a nível nacional, a atividade da medicina transfusional e da transplantação e garantir a dádiva, colheita, análise, processamento, preservação, armazenamento e distribuição de sangue humano, de componentes sanguíneos, de órgãos, tecidos e células de origem humana.

As comunicações entre os serviços são realizadas através do protocolo HL7 (*Health Level 7*), um protocolo para troca de mensagens entre os diferentes serviços da saúde, definem um conjunto de normas internacionais para a representação e a transferência de dados clínicos e administrativos entre sistemas de informação em saúde, como clínicas, consultórios, hospitais, sistemas de saúde pública, entre outros. Este protocolo é utilizado para permitir que as organizações de saúde partilhem facilmente informações clínicas. As integrações do *Clinidata®Transfusion* com outros serviços não estão dentro do âmbito deste PEI, tendo sido desenvolvido por programadores seniores. [15] Uma das maneiras utilizadas para garantir a identificação e distribuição das unidades com eficácia, precisão, segurança para o benefício dos dadores e pacientes em todo mundo, passa pela impressão de etiquetas para as unidades segundo as normas do IBST128. Trata-se de um padrão global para identificação das unidades sanguíneas, que define os campos a serem mostrados nas etiquetas. [30].

2.2 Gestão de Projeto

Nesta seção é descrito o processo de gestão do *sw Clinidata®*, na qual o módulo *Clinidata®Transfusion* está integrado.

Segundo *Pressman* [16] a gestão de projetos de software é descrita pelos “4 P’s” e a sua ordem não é arbitrária, focando-se nas **P**essoas, **P**roduto, **P**rocesso e **P**rojecto

2.2.1 As Pessoas

As funções e as responsabilidades de cada elemento da equipa do projeto estão definidas no manual de funções e responsabilidades da Maxdata.

No âmbito deste PEI, salientam-se os seguintes:

- **Gestor de projeto:** responsável pela estruturação e planeamento dos projetos, monitoriza, avalia e realiza o fecho do projeto;
- **Diretor de Investigação e desenvolvimento:** responsável pela equipa de I&D (Investigação e desenvolvimento), gere os circuitos de desenvolvimento do software, realiza a validação da análise funcional do sistema e avalia os analistas/programadores;
- **Analistas seniores:** responsáveis pela prestação de serviços de consultadoria no cliente, efetuam o levantamento de requisitos e análise funcional do software. É formado por colaboradores com vários anos de experiência e são eles que melhor conhecem as necessidades do utilizador final;
- **Técnicos de assistência:** registam, avaliam, resolvem incidentes e pedidos de serviço em visitas aos clientes. Responsáveis pelos testes de software, instalam *releases* (é a produção de uma versão final da aplicação) e integram os equipamentos;
- **Gestor da Segurança da Informação:** responsável por toda a gestão de segurança da informação da aplicação e se está de acordo com as normas exigidas;
- **Code reviewer:** responsáveis pela revisão do código desenvolvido no código fonte do *software* da Maxdata;
- **Utilizadores:** funcionários da saúde que trabalham com a aplicação diariamente, nomeadamente médicos, responsáveis por interagir com a aplicação para consulta e requisição de unidades transfusionais, enfermeiros para a etapa de colheita e monitorização do paciente e técnicos laboratoriais responsáveis por registar o resultado dos exames, decompor, validar e processar as unidades sanguíneas.
- **Cliente:** ajudam a definir requisitos, o cliente neste contexto corresponde a um conjunto de hospitais e laboratórios clínicos onde a Maxdata já tem os seus produtos instalados e foram o ponto de partida para o arranque da nova aplicação;
- **Programadores/Analistas:** efetuam análises funcionais de *sw* de acordo com a metodologia definida no documento de análise para cada módulo que é integrado no *Clinidata®*, tendo sido a minha função no decorrer deste PEI. Tive como responsabilidade a implementação em falta dos ecrãs de transfusão e a implementação do módulo de dadores: criação dos domínios, serviço, repositório e todos os ecrãs inerentes ao módulo. O desenvolvimento foi feito segundo as metodologias definidas no documento de análise.

2.2.2 O Produto

O *Clinidata®Transfusion* é um sistema de gestão de dadores, transfusões e banco de sangue, e tal como todo o *Clinidata®*, segue um modelo cliente-servidor, é uma aplicação *web* implementada em linguagem *Java*, tanto no *back-end* como *front-end*, através da ferramenta de desenvolvimento GWT⁶ (*Google Web Toolkit* – ver seção 4.1). Assim, no lado do cliente é executada pelo browser em *JavaScript*, a conversão do código de *Java* para *JavaScript* é da responsabilidade do *GWT* e no lado do servidor é utilizado as *frameworks Spring* e *Hibernate*, para tratar das diferentes bases de dados existentes e o *LiquidBase* para as alterações de estrutura na base de dados que são realizadas. Funciona em *cloud computing* e suporta as principais *SGBDS* (*Oracle*, *SQL Server*, *PostgreSQL*, *MySQL*). A arquitetura é orientada a serviços *SOA* (*Service Oriented Architecture*) que garante comportamentos coerentes e fiáveis nos vários cenários de utilização, facilitando a interoperabilidade com outros sistemas. Permite o rastreio das operações de acesso, inserção, alteração e eliminação de dados pessoais, proteção da privacidade, incluindo anonimização/cifra dos dados pessoais na BD e utilização de técnicas avançadas como a criptografia baseada em atributos.

No contexto deste PEI, fui integrada na implementação do *Clinidata® Transfusion* que é composto por três -módulos:

- **Gestão de doações**, abrange todo o processo desde a consulta do dador, colheita, decomposição das unidades colhidas até à validação das unidades finais para ficarem disponíveis para a transfusão,
- **Gestão de transfusões**, abrange todo o processo desde o pedido de transfusão pelo médico no serviço, até à administração de unidade(s) de sangue compatível com o grupo sanguíneo do paciente, medições vitais incluindo o registo e investigação de possíveis reações adversas,
- **Gestão de unidades**, contempla a troca de unidades sanguíneas com entidades externas (envio e receção).

O trabalho descrito neste relatório vai incidir maioritariamente no módulo de dadores.

2.2.3 O Processo

A metodologia de desenvolvimento utilizada na Maxdata e consequentemente neste PEI, é baseada em metodologias de desenvolvimento *Agile*⁷, sendo constituído por um

⁶ GWT: <http://www.gwtproject.org/overview.html>.

⁷ Manifesto for Agile Software development, <http://agilemanifesto.org/iso/en/manifesto.html>.

conjunto de iterações de curta duração, aplicadas sequencialmente às funcionalidades a desenvolver. Cada iteração é composta pelas fases de levantamento de requisitos, análise e desenho, codificação, testes e documentação, como ilustrado na figura 7.

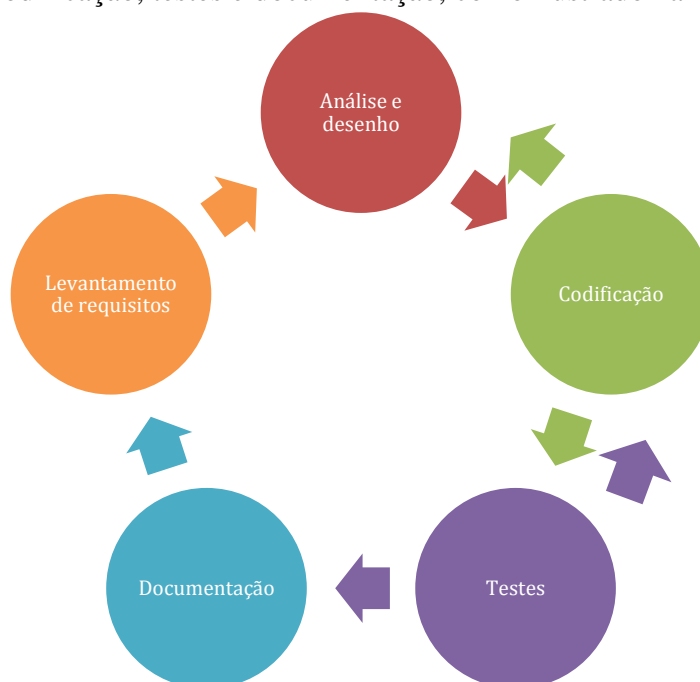


Figura 7 – Iterações no modelo de desenvolvimento ágil usado na Maxdata

Na primeira fase, o **levantamento de requisitos**, são definidos todos os requisitos das funcionalidades a desenvolver. Na análise e detalhe da funcionalidade pode ser necessário realizar reuniões com os clientes para detalhar os processos de trabalho e regras de negócio a seguir, analisar a legislação a aplicar e quais as regras a seguir.

Na fase de **análise e desenho**, realizada pelos analistas seniores da Maxdata, define-se a forma mais correta de implementar os requisitos recolhidos na fase anterior, assegurando a sua compatibilidade e coerência com as funcionalidades já existentes. Esta fase tem como *input* os requisitos funcionais e não funcionais da aplicação e a funcionalidade a desenvolver. O *output* consiste na elaboração de documento de análise por cada modulo a desenvolver, que detalha e define o modelo de dados, tabelas a criar, campos a serem criados, protótipos de baixa fidelidade dos ecrãs e toda a lógica de negócio. Todos os módulos são implementados pela equipa de I&D seguindo o documento de análise durante a fase de codificação.

Na fase de **codificação** é feita a implementação das funcionalidades utilizando as técnicas e linguagens de programação mais adequadas, assegurando a correção dos erros que eventualmente tenham sido detetados em testes anteriores. Esta fase pode conter dois tipos de *input*: documento de análise e desenho da funcionalidade a desenvolver ou uma lista de erros detetados durante os testes efetuados à funcionalidade. Se surgir algum impedimento na codificação na funcionalidade, a

mesma regressa à fase de análise e desenho, prosseguindo novamente para esta fase, mas com o documento de análise e desenho devidamente atualizado. O processo de implementação da funcionalidade é realizado pela equipa de I&D e o código produzido é integrado num sistema central de gestão de versões, o *GIT*⁸. Diariamente, é enviado para toda a equipa de desenvolvimento um relatório automático com a atividade de cada elemento e no seu seguimento é realizada uma *daily standup meeting* [17] onde é feito um acompanhamento dos trabalhos realizados por cada elemento da equipa de I&D. Nesta fase são também efetuadas revisões ao código [18] que ocorrem após a codificação de uma determinada funcionalidade e que são efetuadas por um dos programadores seniores (*code reviewers*), que analisam o código em conjunto com o programador que o desenvolveu. Estas revisões encorajam a partilha de conhecimento, são encontrados bugs durante o processo, forçam a escrita de código mais legível (para ser entendido mais facilmente), entre outras vantagens [19].

Na fase de **testes**, são efetuados pelos colaboradores que não foram responsáveis pela implementação da funcionalidade. O seu objetivo é testar a funcionalidade implementada na etapa de codificação e num ambiente que simula as diversas realidades dos clientes (diferentes *browsers*, diferentes SGBDs), de modo a verificar se o comportamento da aplicação é idêntico nos diferentes cenários e contextos. Quando são encontrados problemas, o diretor de I&D revê os resultados dos testes e decide se a funcionalidade deve ou não retornar para a fase de codificação.

Para finalizar, na fase de **documentação** é atualizada a documentação técnica e funcional da aplicação. Após os testes, a equipa de testes e/ou marketing avalia as funcionalidades que foram implementadas e atualiza a documentação com a informação necessária.

2.2.4 O Projeto

Este projeto foi dividido em três fases:

- **Fase 1 (Setembro a Dezembro de 2018)**

Esta fase iniciou-se com a leitura de documentação sobre as tecnologias usadas no *Clinidata*®, o estado atual do mesmo, leitura do documento de análise do banco de sangue e elaboração do relatório preliminar.

Quando entrei na Maxdata, estavam a terminar os módulos de transfusão e unidades do *Clinidata*® *Transfusion*. Após a leitura da documentação, foram-me atribuídas pequenas tarefas de desenvolvimento no módulo *Clinidata*® *Transfusion*,

⁸ GIT: www.git.com

como algumas correções ou acrescento de algum campo nos ecrãs já desenvolvidos. Existiu uma forte componente prática nesta fase que foram fundamentais para a minha integração de desenvolvimento da aplicação, uma vez que me deram a conhecer os detalhes da arquitetura da aplicação e sobre os vários conceitos da lógica de negócio.

No dia 30 de outubro de 2018 obtive a formação no Hospital de Santa Maria, onde foi possível observar o funcionamento de um laboratório de análises clínicas e o software *Clinidata®* a ser utilizado por utilizadores reais.

- **Fase 2 (Janeiro 2019 a Junho de 2019)**

Nesta fase foi realizada a conceção e desenvolvimento do módulo de gestão de dadores e implementação dos ecrãs “Gestão de Transfusões”, “Detalhe de transfusão” e “Medições vitais”. Para tal, foi necessário:

- (i) desenvolvimento na camada de dados - criação das tabelas de dados e modelos de domínio inerentes ao módulo de dados;
- (ii) adicionar à geradora do banco de sangue as novas funcionalidades (Gestão de transfusões, detalhe de transfusão, Gestão de doações, detalhe de doação e medições vitais) e criação das diferentes *labels* para os novos campos a serem exibidos nos novos ecrãs;
- (iii) desenvolvimento na camada de negócio - criação do *service* “DonationBC” e do *repository* “DonationDAO”
- (iv) desenvolvimento da camada de cliente - implementação dos ecrãs seguindo um modelo MVP e respeitando as metodologias do documento de análise do banco de sangue – “Gestão de doações”, “Detalhe de doação”, “Validação de unidades”, “Decomposição de unidades”, “Medições vitais”, “Gestão de transfusões” e “Detalhe de transfusão”.
- (v) correções/ alterações ao módulo - fui responsável por corrigir erros reportados do *Clinidata® Transfusion*, consoante o que foi reportado pela equipa de testes ou alterações que foram pedidas pelos clientes no módulo de transfusões e correções *pós-review* ao módulo desenvolvido de dados.

O *Clinidata®* já tinha um módulo de consultas e colheitas, pelo que foram necessárias várias alterações aos módulos para passar a funcionar de acordo com o novo módulo de doações, como por exemplo, as consultas podiam ser de dois tipos: exames ou transfusões, passou a ser de três: exames, transfusões ou doações. Estas alterações aos módulos existentes não foram realizadas por mim. Na colheita de sangue de dador, passou-se a chamar o meu ecrã de medições vitais.

- **Fase 3 (Maio a Julho de 2019)**

Elaboração do relatório final. Na página seguinte, encontra-se disponível o mapa de *gantt* com as devidas tarefas realizadas (Figura 8).

Mapa de Gantt

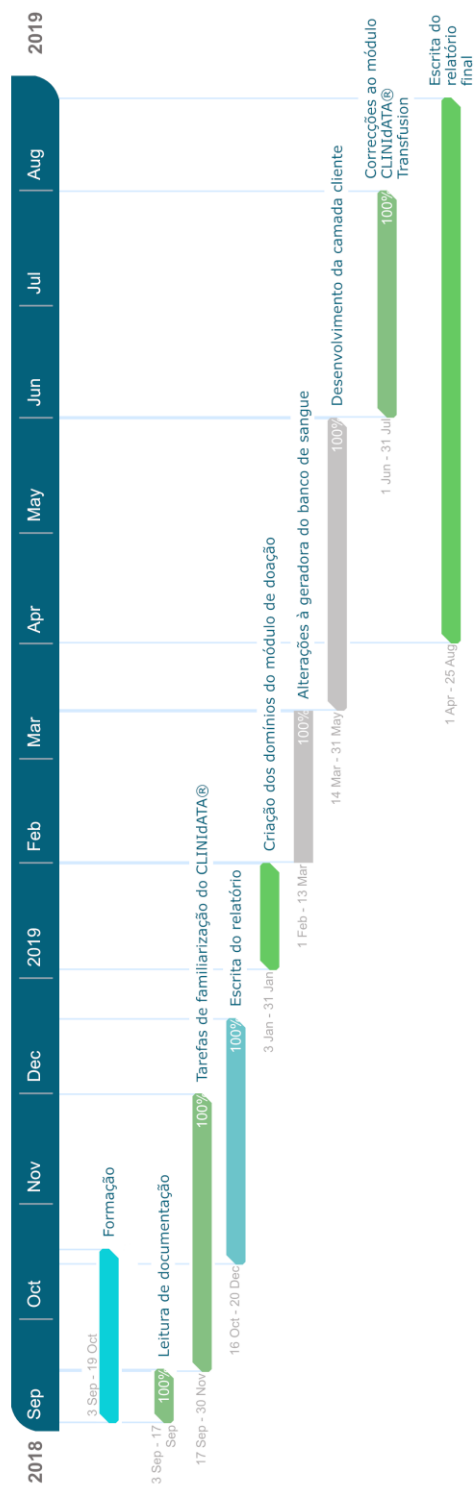


Figura 8. Mapa de Gantt

2.3 Trabalho relacionado

Nesta seção são apresentados alguns sistemas de gestão de informação para o processo transfusional na área da saúde que competem com o *Clinidata@Transfusion*, suportando funcionalidades de gerenciamento de transfusões e bancos de sangue.

Sunquest Blood Bank

A aplicação *Sunquest Blood Bank*¹⁰ é um sistema de gestão de informação de banco de sangue e de transfusões, implementado para atender o fluxo de trabalho do processo transfusional. Permite aos enfermeiros/técnicos de laboratório o rastreo e a gestão dos produtos sanguíneos disponíveis em stock e/ou um serviço de gestão transfusões. Funciona sobre uma arquitetura Cliente-Servidor, base de dados relacional (*Oracle®*) e sistemas operativos *Linux* e *Windows*. Principais funcionalidades:

- Leitura via *barcode* das unidades sanguíneas processadas que vão seguir para ser transfundida a um paciente, através das etiquetas que seguem a norma ISBT 128.
- Os estados das unidades são atualizados automaticamente durante todas as etapas.
- Visualização dos resultados dos exames realizadas as unidades sanguíneas que vão seguir para transfusão.
- Guarda o registo dos pacientes permitindo o rastreo dos seus anticorpos, número de unidades transfundidas, data da última transfusão, grupo abo Rh. [20].

The screenshot displays the Sunquest Blood Bank software interface. At the top, it shows patient information: 112112, WEL, CHARLOTTE, BMS No. XXXXXXXXX, DOB 01/01/1969 (40Y), Sex F, and HED: REDI. Below this, there are tabs for Antigens/Antibodies, Problems, Comments, and Transfusion. The main section is titled 'CROSSMATCH (WITH %EXX)' and shows specimen testing results. It includes a table with columns for Test, Description, and Result. The results show various tests like ABO/RH(D), ANTI-BODY SCREEN SUMMARY, and CROSSMATCH EXPIRATION, all with results like POSITIVE, NEGATIVE, or PEND. At the bottom, there is a section for Compatibility Testing, showing a table with columns for Unit, XM, YS, and CROSSMATCH RESULT. The interface also includes buttons for 'Accept', 'Cancel', and 'Interp'.

Figura 9 – Exemplo de um ecrã *Sunquest Blood Bank*¹⁰(retirado de ¹⁰)

¹⁰ <http://www.clpmag.com/2016/05/software-release-includes-improved-blood-banking-tools/>

^{TD}BloodBank

A aplicação ^{TD}*BloodBank*¹¹ é um sistema de gestão de informação que gere o fluxo transfusional, nomeadamente entrada de unidades, recepção das amostras e testes, *crossmatch*, ecrã de anticorpos e rastreio pós-transfusional. Principais funcionalidades:

- Gestão de Unidades: Etiquetas das unidades sanguíneas segundo o padrão *ISBT 128*, componentes sanguíneos e derivados, gestão de *stock*, rastreabilidade do local da unidade
- Gestão de Transfusões: Testes de amostras do paciente, testes de compatibilidade, comparação automática com o resultado anterior, histórico transfusional do paciente, transfusão direta e autóloga, requer confirmação dupla dos resultados por parte dos utentes da saúde os resultados dos testes.

Esta aplicação utiliza: uma arquitetura Cliente-Servidor, base de dados relacional (*Oracle®* ou *SQL Server®*), funciona sobre o sistema operativo *Windows®*, comunicações usando o padrão HL7, interface com os tipos sanguíneos e os *crossmatch* [21].

Hemovida

A aplicação Hemovida é um sistema de gestão de informação de bancos de sangue vocacionado para profissionais de saúde, com o objetivo de informatizar todo o ciclo de doação de sangue, desde a colheita até a distribuição da unidade, oferecendo total rastreio de cada etapa do processo. Principais funcionalidades:

- possibilita a inscrição de doadores, dados sobre histórico de doenças, transfusões de sangue, pré-triagem, colheita, fracionamento, imuno-hematologia, sorologia e distribuição
- informa sobre a colheita e processamento do sangue, distribuição, transfusão e descarte de componentes sanguíneos
- Identifica por código de barras as unidades colhidas, processadas e distribuídas

Funciona sobre o sistema operativo *Windows 98* ou superior, utiliza uma arquitetura cliente-servidor, base de dados relacional (*SQL Server®*). Tem como objetivo informatizar todo o ciclo de doação de sangue, desde a colheita até a distribuição das unidades, controlando cada etapa do processo [22].

¹¹ <https://www.technidata-web.com/en-gb/tdbloodbank-en-gb>

SBS.WEB

O *SBS Web* é um sistema de gestão de bancos de sangue e gestão transfusionais, agora desenvolvido na versão *web*. Possui uma plataforma flexível que pode ser configurada de acordo com as particularidades de cada realidade de negócio. Principais funcionalidades:

- Usada a codificação ISBT128 nas etiquetas das unidades
- Identificação de dadores por meio de foto digital e/ou leitura da impressão digital
- Colheita externa informatizada
- Rastreabilidade total das informações sobre as doações e transfusões

Sendo *web*, funciona sobre qualquer o sistema operativo com browser. BDs: Oracle, MySQL, PostgreSQL e Microsoft SQL server [23].

ASIS

A aplicação **ASIS** (Aplicação de um Sistema de Serviço a Serviços de Sangue), é um sistema de informação e base de dados português para os profissionais de saúde da área dos serviços de sangue, que permite a gestão de Serviços de Sangue, Serviço de Medicina Transfusional e Centros de Sangue e Transplantação. Principais funcionalidades:

- Permite gerir todas as atividades relacionadas com a promoção e colheita de sangue (planeamento de colheitas de dádivas, registo de dádivas e de dados clínicos de dadores)
- Processamento (separação de componentes sanguíneos, realização de análises laboratoriais, qualificação das dádivas)
- Integração de resultados analíticos processados a partir de amostras
- Validação de componentes e finalmente a sua distribuição pelos hospitais do país.
- Permite aos Serviços de Medicina Transfusional registar a efetivação da transfusão, visualizar os resultados laboratoriais do doente e os registos de transfusão de componentes e produtos sanguíneos.
- Pode integrar com o sistema de informação de Gestão Administrativa do Hospital SONHO

.O ASIS funciona sobre o sistema operativo *Windows*, utiliza *SGBDR Oracle* e está instalado em vários serviços de sangue e nos Centros de Sangue e Transplantação do IPST, engloba também uma base de dados nacional de monitorização de reservas de sangue [23]

Comparações

Na tabela em seguida, é apresentada uma suma de cada aplicação referida anteriormente e do *Clinidata®Transfusion*: os serviços que dispõem, os tipos de SGBDR e o seu ambiente de execução.

Aplicação	Gestão de doações	Gestão de transfusões	Gestão de Unidades	SQL Server	PostgreSQL	Oracle Database	Windows	Linux
<i>Sunquest Blood Bank</i>		x	x			x	x	x
<i>TDBloodBank</i>		x	x	x		x	x	
<i>Hemovida</i>	x		x	x			x	
<i>SBS Web</i>	x	x		x	x	x	x	x
<i>ASIS</i>								
<i>Clinidata® Transfusion</i>	x	x	x	x	x	x	x	x

Tabela 2 – Comparação entre as diferentes aplicações transfusionais

Como é possível observar pela Tabela 2, só o *Clinidata®Transfusion*, oferece os três serviços (doações, transfusões e unidades). Em relação ao tipo de SGBDR, todas as aplicações apresentadas com exceção do *SBS.Web* e o *Clinidata®Transfusion*, estão limitadas a um ou dois tipos de base de dados, restringindo bastante os clientes que podem usufruir deste serviço. O *SBS.WEB* e o *Clinidata* distinguem-se positivamente das restantes pelo facto de ser um modelo SaaS (Software-as-a-service), ou seja, permite ao cliente utilizar o software apenas com uma ligação à internet, não sendo necessário uma aquisição de servidores e de licenças. Por ser *web*, permite que a aplicação seja executada em qualquer sistema operativo com acesso a um browser, ao contrário das restantes aplicações que estão limitadas à execução em ambiente *Windows*.

Assim podemos afirmar que o *Clinidata®Transfusion* é a aplicação mais completa oferecendo uma gestão de dadores, transfusões e unidades. Funciona sobre os SGBDR *SQL Server*, *PostgreSQL* e *Oracle Database* e sendo uma aplicação *web*, executa nos browsers mais comuns, suportando qualquer sistema operativo com acesso a um browser. Pode verificar-se que a maior parte das aplicações executam apenas em ambientes *Windows*, mas a tendência será para executarem em ambiente *web*, para que possam assim ser utilizadas por qualquer dispositivo móvel com acesso a um *browser*, permitindo aos utilizadores o acesso à informação *on-the-go*.

Capítulo 3 Análise de requisitos

Neste capítulo são detalhados os requisitos, modelo de dados e lógica de negócio através de diagramas para a implementação do módulo de dadores, obtidos a partir do documento de análise do banco de sangue.

Na Maxdata, para cada novo módulo a implementar e integrar no software *Clinidata®*, é feito um levantamento de requisitos e análise funcional do software pelos analistas seniores pois são eles os responsáveis pela prestação de serviços de consultadoria no cliente. Posteriormente é realizado um documento de análise que detalha e define os requisitos funcionais, modelo de dados, tabelas a criar, os campos a serem criados, protótipos de baixa fidelidade dos ecrãs e toda a lógica de negócio associada ao módulo, assegurando a sua compatibilidade e coerência com as funcionalidades já existentes. Quando fui integrada ao *Clinidata®Transfusion*, os requisitos inerentes às transfusões e unidades já estavam implementados, levando a que o foco do meu PEI seja o módulo de dadores.

3.1 Requisitos

Pode definir-se como requisito uma funcionalidade/comportamento a ser contemplada pela aplicação. O levantamento de requisitos é uma fase fundamental no ciclo de vida de *sw*, pois é a partir da sua especificação que permite que a aplicação vá de encontro com as necessidades de negócio. Na Maxdata, os requisitos são classificados como obrigatórios ou opcionais e a sua origem poderá ser do cliente (através de pedidos ou necessidades do cliente) ou poderá ser interna (estratégia definida pela empresa). Foi feita a distinção dos requisitos por dois tipos:

- **Requisitos funcionais:** descrevem os comportamentos que o sistema deve ter, refletindo os objetivos estabelecidos para a aplicação durante as reuniões com o cliente, como por exemplo funcionalidades específicas da aplicação e níveis de desempenho definidos.
- **Requisitos não funcionais:** descrevem restrições que devem ser respeitadas durante o desenvolvimento e requisitos que estão implícitos na aplicação

3.1.1 Requisitos Funcionais

Na tabela seguinte encontram-se identificados os principais requisitos funcionais do módulo de dadores desenvolvido, tendo sido responsável por implementar os requisitos que se encontram com um “*”.

Requisito	Operação associada	Descrição
RF1	Consulta de doação	Categorizar o tipo de doação (autóloga, homóloga ou sangria)
RF2	Consulta de doação	Se a consulta a doação for rejeitada, o dador encontra-se impedido de fazer doações, devendo o sistema deve bloquear o registo da consulta.
RF3	Consulta de doação	Se a pré-triagem foi aprovada, o dador preenche um questionário.
RF4	Consulta de doação	Se o dador não teve nenhum impedimento, a doação passa a ficar com estado “pendente” (de colheita)
RF5	Consulta de doação	O médico insere quais os exames a realizar e insere os componentes que se quer produzir a partir da dádiva (requisição de componentes).
RF6	Consulta de doação	É atribuído um número de doação único composto por ano + hospital_id + nºsequencial
RF7	Consulta de doação	Requisição do exame de teste para descobrir o tipo de AB0 e Rh do dador e exames de despistes
RF8	Colheita	Identificação do kit que vai armazenar a dádiva
RF9*	Medições vitais	Antes de iniciar a colheita são registados os sinais vitais e respetiva data e hora.
RF10*	Medições vitais	Realizar medição vital antes e no fim da colheita
RF11*	Medições vitais	Quando é realizada a medição final de uma doação, a doação passa para estado “Realizada” (colheita)
RF12*	Medições vitais	Após a colheita é necessário pesar o saco colhido e introduzir o valor em ml, sendo obrigatório registar o volume colhido na medição vital “final”
RF13*	Medições vitais	Identificar se ocorreu alguma reação adversa

RF14*	Medições vitais	Em caso de ter ocorrido uma reação adversa, escolher uma ação a tomar
RF15*	Medições vitais	A medição vital tem que ter uma data superior de 15 minutos em relação à anterior
RF16*	Medições vitais	As medições vitais vão ser necessárias nas transfusões e nas doações, a sua pesquisa e introdução vai ser feita através do código de doação no caso de ser uma doação, ou do número de transfusão e número de unidade no caso de uma transfusão
RF17*	Decomposição de unidades	Consultar/alterar/adicionar/eliminar as componentes a gerar a partir da doação.
RF18*	Decomposição de unidades	Criar o registo automático das unidades decompostas a partir dos componentes requisitados na consulta de doação
RF19*	Decomposição de unidades	As unidades decompostas ficam com estado “pendente” (dos resultados dos exames requeridos na consulta de doação para serem validadas ou rejeitadas)
RF20*	Decomposição de unidades	Quando existe a inserção de uma unidade decomposta, a doação passa para estado “decomposta”
RF21*	Decomposição de unidades	Cada doação só produz uma unidade de cada componente
RF22*	Decomposição de unidades	Deve ser possível alterar/adicionar/eliminar as componentes a partir de uma doação
RF23*	Decomposição de unidades	Gerar etiquetas de acordo com a norma ISBT 128 para identificar cada uma das unidades a gerar.
RF24*	Validação de unidades	Aprova ou rejeita as unidades após a realização das análises. Consulta, aprova ou rejeita as unidades da doação após a realização das análises.
RF25*	Validação de unidades	Devem ser aplicadas algumas regras automáticas que alertem o utilizador ou proíbam a validação em função ao resultado das análises.
RF26*	Validação de unidades	Alteração dos estados das unidades doadas para “validadas” ou “canceladas” tendo em conta os resultados dos exames, ou rejeitar.
RF27*	Validação de unidades	Quando uma unidade é registada como validada passa para estado “validada”

RF28*	Validação de unidades	Quando uma unidade é registada como cancelada passa para o estado “cancelado”
RF29*	Validação de unidades	Quando uma unidade doada é registada como validada e a doação não tem unidades com estado “pendente”, a doação passa para o estado “validada”.
RF30*	Validação de unidades	Quando uma unidade doada é registada como cancelada e a doação não tem unidades pendentes nem validadas, a doação passa para o estado “cancelada”
RF31*	Validação de unidades	As unidades registadas como canceladas, é obrigatório o preenchimento do seu motivo de cancelamento*
RF32*	Validação de unidades	Após validação de unidades, deve ser impressa etiquetas finais para identificação da unidade de acordo com a norma IBST128.
RF33*	Validação de unidades	Se a unidade foi validada, é transferida para um local onde estão armazenadas as restantes unidades disponíveis para administração.
RF34*	Dador, doação e componentes	Identificador único

Tabela 3. Requisitos funcionais associados ao módulo dadores do Clinidata®*Transfusion*

3.1.2 Requisitos Não funcionais

Os requisitos não funcionais deste módulo, bem como em todos os módulos presentes na aplicação *Clinidata®* definem qualidades globais ou atributos do sistema, nomeadamente:

- *Cross browser*, ou seja, ter um comportamento similar ao ser executado nos *browsers* mais comuns,¹² incluindo *Internet Explorer*, *Google Chrome*, *Mozilla Firefox* e *Apple Safari*;
- *Cross-platform*, independente do SGBD utilizado, uma vez que diferentes clientes optam por BDs diferentes. Assim poderão executar sobre os SGBDs mais comuns incluindo *Oracle DataBase*, *Microsoft Sql Server*, *PostGreSQL* e *MySQL*.
- Experiência de utilizador, a aplicação conta com diferentes tipos de utilizadores, uns com mais experiência e, de acordo com Sainfort et al. [26]¹³ os principais

¹² Browser statistics, http://www.w3schools.com/browsers/browsers_stats.asp

¹³ Capítulo 33, “Human-Computer Interaction In HealthCare”

desafios das aplicações de sw na área da saúde passam pela personalização e adaptação, por isso é importante criar aplicações que sejam configuráveis, para de modo a responder facilmente às diversas preferências dos vários perfis de utilizador.¹⁴ Como tal, a aplicação é configurável de diversas maneiras: é possível mudar o tema da aplicação; as disposições dos campos nos ecrãs são editáveis, incluir um ou não um campo, permite ao utilizador editar a ordem de determinados campos consoante a sua preferência, entre outros.

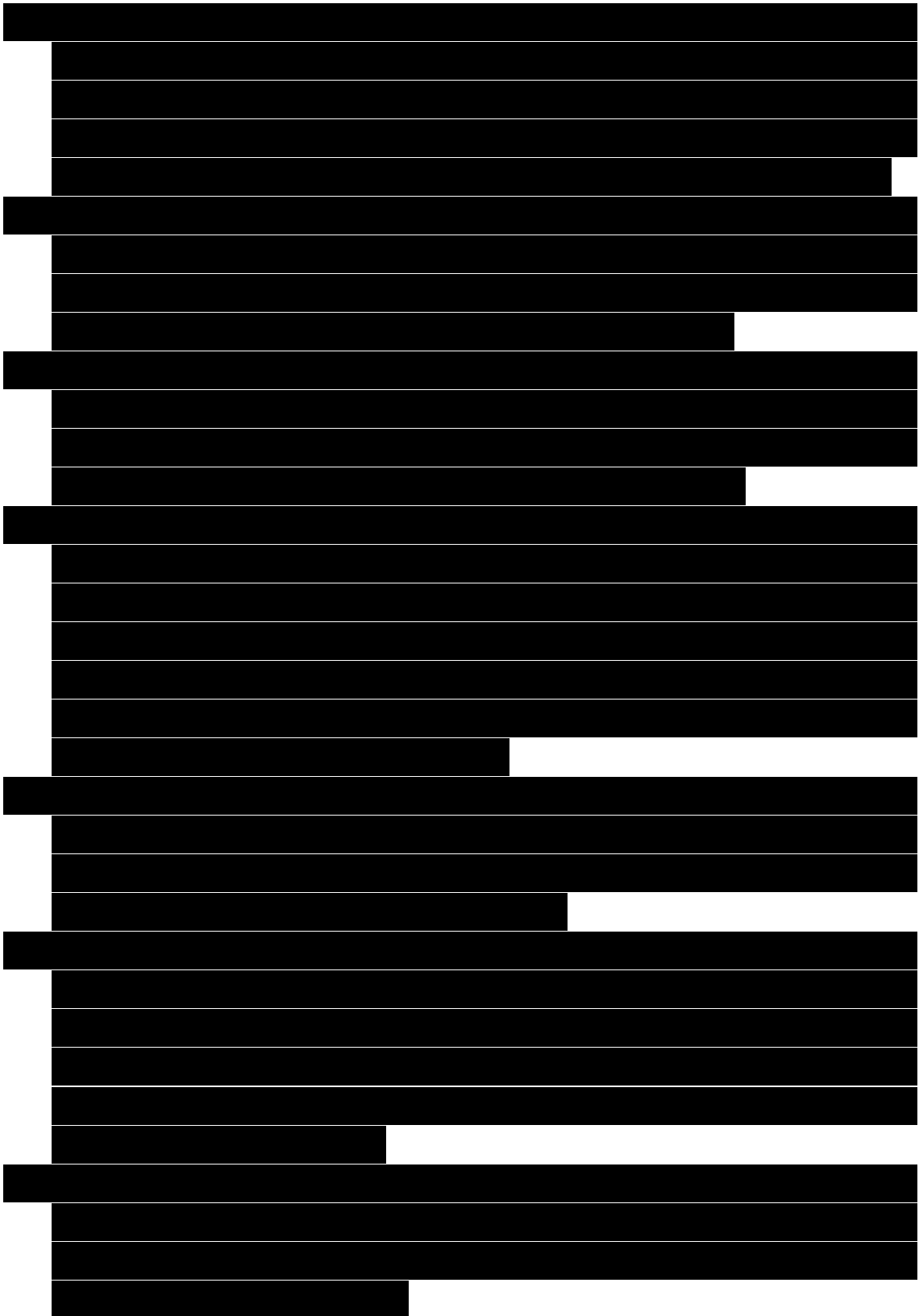
- Independente do idioma de desenvolvimento, é possível traduzir toda a aplicação para um idioma diferente do original sem ter de alterar o código da aplicação.
- Desempenho, é um requisito fundamental da aplicação pois serão manipuladas quantidades consideráveis de dados e, deste modo, é esperado que as pesquisas e as operações sobre este modelo de dados deverão ser rápidas e eficazes.
- Usabilidade, a interface deve ter uma interface simples e intuitiva de modo a beneficiar a experiência do utilizador. Sendo uma aplicação *web*, impõe-se a compatibilidade com os *browsers* mais comuns.
- Fiabilidade, a informação gerida pelo sistema e consistência dos dados. A segurança dos dados é também importante de modo a proteger as informações contra acessos não autorizados.
- Manutenção, o desenvolvimento da aplicação deve ser feito tendo em conta futuras atualizações
- Portabilidade, funciona para qualquer dispositivo móvel com acesso a um *browser*
- Integridade, as informações apresentadas dependem do tipo de utilizador do *Clinidata*®

3.2 Modelo de Dados

O *Clinidata*® segue um modelo de dados relacional e os seus dados podem ser armazenados em diferentes SGBDR (*Oracle DataBase*, *Microsoft Sql Server*, *PostGreSQL* e *MySQL*). As tabelas do modelo relacional da aplicação encontram-se nomeadas do seguinte modo:

- **Tabelas de configuração:** tipicamente estas tabelas têm um tamanho fixo e representam tabelas que não crescem dinamicamente. Este tipo de tabela é identificado com o prefixo “T” no seu nome, seguindo com duas siglas que representam o nome do módulo que está integrada e por fim, o modelo de domínio que representam. Armazenam toda a informação relacionada com a configuração

The image consists of a single, uniform black rectangle that fills the entire frame. There are no discernible features, text, or patterns other than the solid black color.



Na figura 10 é possível observar uma versão resumida do modelo de domínio do módulo de dadores que só ilustra as chaves primárias e chaves secundárias de cada tabela. A versão completa do modelo de domínio encontra-se na secção 7.1 (Anexos).

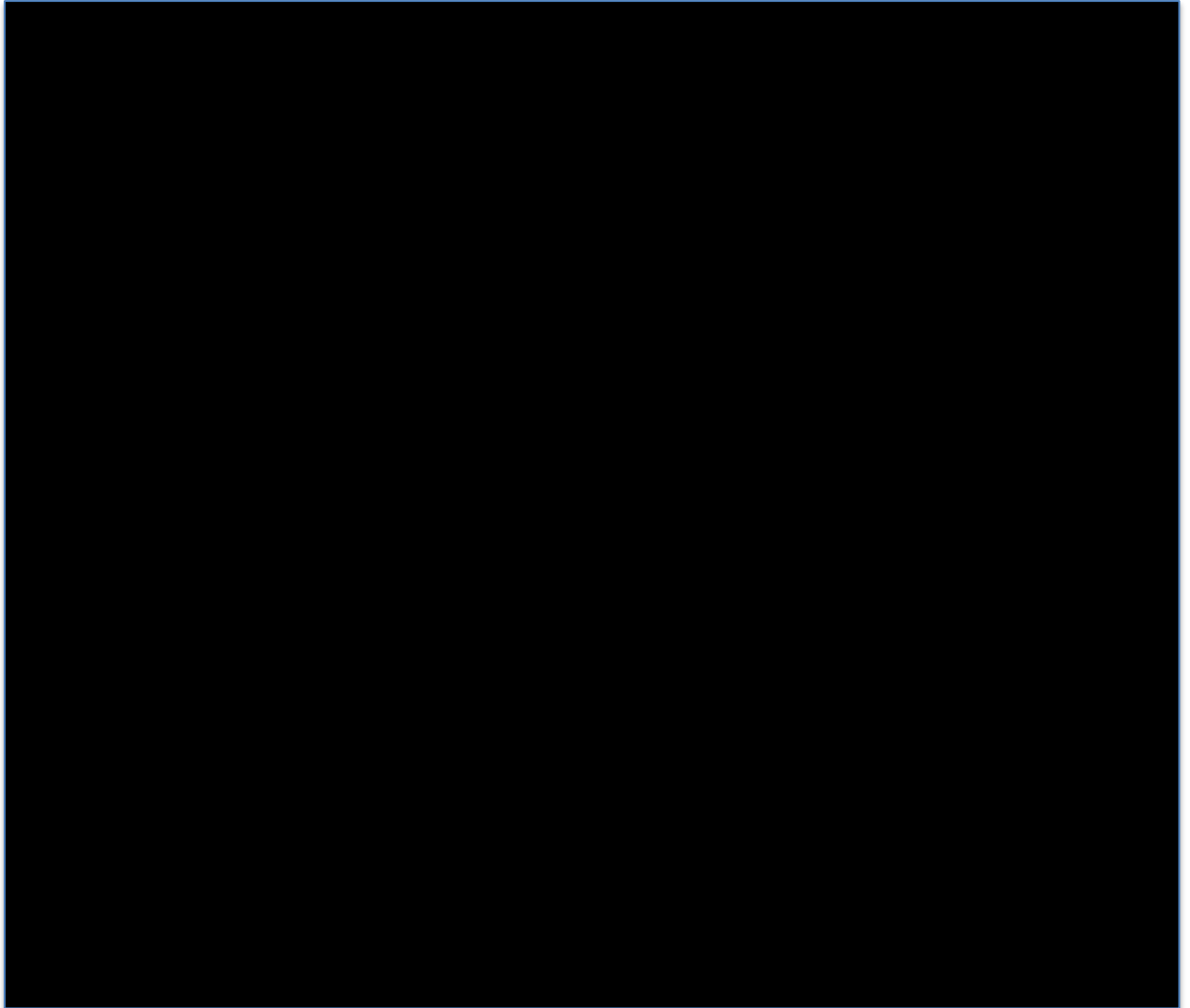
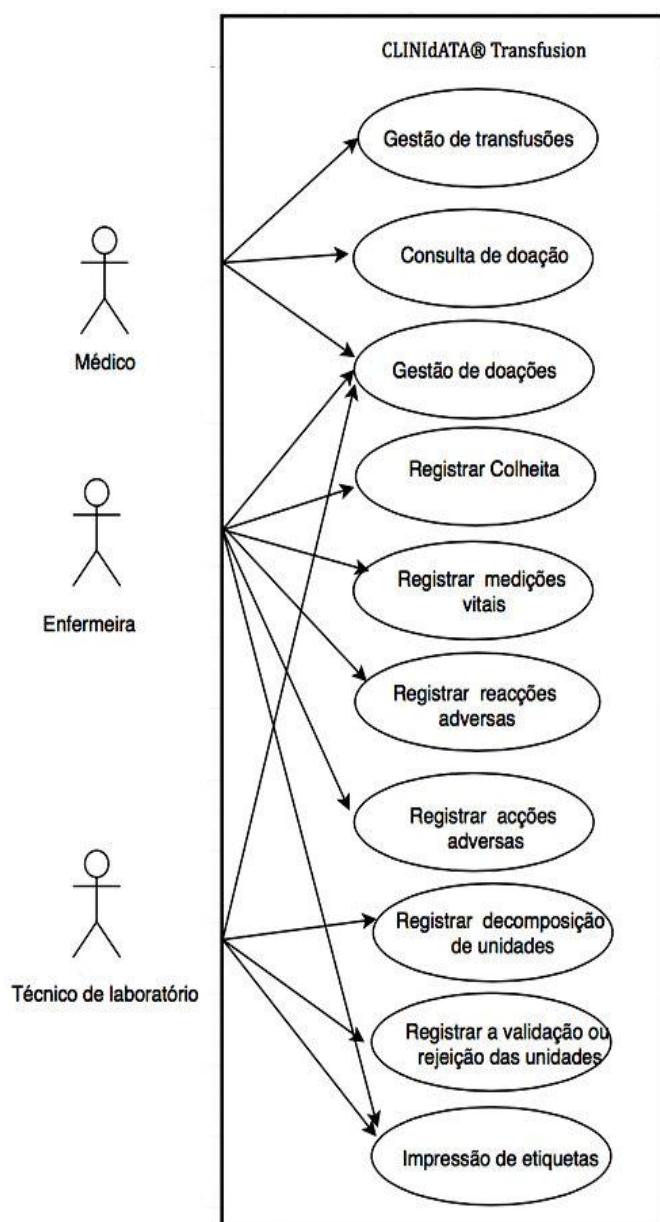


Figura 10. Modelo de domínio do módulo de dadores

3.3 Diagrama de casos de uso e de fluxo

Aqui é apresentado um diagrama de casos dos ecrãs do módulo de dadores e dos ecrãs de gestão de transfusões e medições vitais implementados durante este PEI.

O diagrama de casos de uso representa as funcionalidades que o módulo implementado teve de satisfazer, e os diferentes tipos de utilizadores envolvidos. Na figura 11 é possível observar as funcionalidades inerentes ao projeto, sendo que há três tipos de utilizadores a interagir com o sistema: o médico, enfermeira e técnico de laboratório.



O **médico** tem acesso à gestão de transfusões, consulta de doação e gestões de doações. É na consulta de doações que o médico indica as unidades a obter a partir da dádiva e os exames a efetuar.

A **enfermeira** está envolvida nas funcionalidades relacionadas com as colheitas de sangue, nomeadamente: registar colheita, medições vitais, ações e reações adversas. Quando é colhida a dádiva, é impresso uma etiqueta para colocar no saco da dádiva.

O **técnico de laboratório** recebe a dádiva, realiza os diferentes testes e exames, decompõem unidades e valida ou rejeita unidades consoante o resultado dos exames. Na fase de decomposição é impresso etiquetas “iniciais” para identificar cada uma das unidades a gerar a partir da dádiva, na fase de validação se a unidade foi validada é impressa etiquetas “finais” para identificação da unidade, prosseguindo para stock para ser disponibilizada para uma ou mais transfusões.

Figura 11. Diagrama de casos de uso do módulo de dadores e da gestão de transfusões (Elaboração própria)

Capítulo 4 Implementação

Neste capítulo são apresentadas as diferentes ferramentas utilizadas para a implementação deste projeto, um fluxo de sequência do módulo de dados, a arquitetura de sistema e o trabalho realizado nas diferentes camadas.

4.1 Ferramentas utilizadas

Neste ponto começo por referenciar as diferentes tecnologias e *frameworks* utilizadas para a implementação deste PEI.

Java

A aplicação foi desenvolvida utilizando a linguagem de programação *Java*¹⁵, tanto *front-end* como *back-end*. Trata-se de uma das tecnologias mais populares e utilizadas em todo o mundo. A utilização de *Java* em *front-end* deve-se à ferramenta GWT, responsável por converter o *Java* em *JavaScript* e HTML para o *browser*, tal como o inverso. Assim, é possível a partilha de código entre o cliente e servidor.

Hibernate

O *Hibernate*¹⁶ é uma biblioteca ORM (*Object Relational Mapping*) desenvolvida para *Java*, que realiza o mapeamento entre os campos presentes nos diferentes objetos de negócio e as tabelas da base de dados, criando assim uma camada transparente de persistência. As diferentes classes *Java* são representadas como tabelas de base de dados e criadas com um esquema que assegura equivalência, refletindo e persistindo as alterações sobre esses objetos (denominados de POJOs¹⁷) na tabela correspondente. Os POJOs, são objetos *Java* que seguem um desenho simplificado em contraposição aos EJBs, por exemplo. Um *JavaBean* é um POJO que segue definições rígidas de estrutura (construtor *default* sem argumentos e métodos que seguem o padrão de *getters* e *setters* para os seus atributos).

O *Hibernate* tem a sua própria linguagem para realizar *queries*, o HQL¹⁸, permitindo a construção de *queries* com uma estrutura semelhante ao SQL, com a diferença que estas são executadas sobre os objetos mapeados pelo *Hibernate*. Este também fornece uma camada de abstração entre a aplicação e a SGBD de forma a que mudanças de SGBD apenas requeiram alterações a nível de configuração, permitindo

¹⁵ Java, http://www.java.com/en/download/whatis_java.jsp.

¹⁶ Hibernate, <http://www.hibernate.org/>.

¹⁷ POJO – Plain Old Java Object, <http://www.martinfowler.com/bliki/POJO.html>

¹⁸ HQL – Hibernate Query Language, <http://docs.jboss.org/hibernate/orm/3.3/reference/en/html/queryhql.html>

que aplicação possa estar disponível para qualquer tipo de SGBD. Esta particularidade é bastante vantajosa uma vez que as aplicações da área de saúde requerem constante adaptabilidade a novos paradigmas de desenvolvimento de software. Também é responsável por facilitar a conexão com a base de dados reduzindo drasticamente o número de linhas de código que são utilizados para as operações CRUD (*Create/Read/Update/Delete*) aos dados. Outra grande vantagem é que gere igualmente de uma forma transparente as operações de conexão/desconexão de cada uma das transações entre a aplicação e a base de dados.

Na figura 12 encontra-se um exemplo de uma HQL realizada para o módulo de dados.

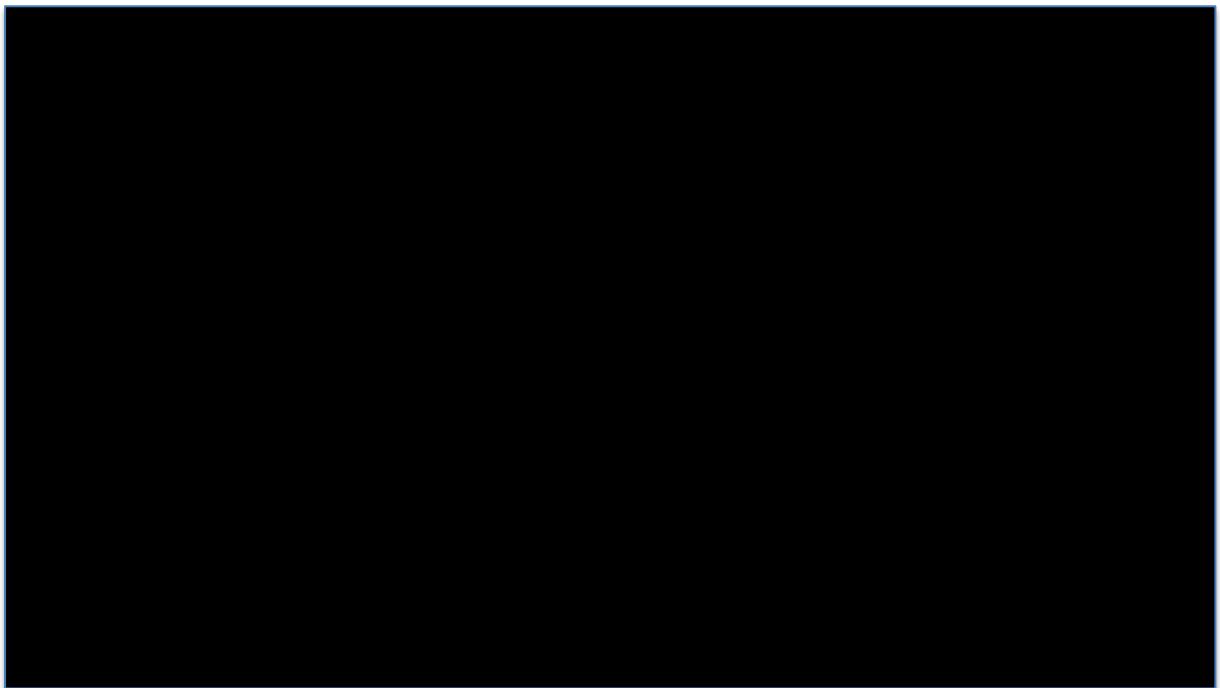


Figura 12. Exemplo de uma HQL Query realizada neste PEI

Spring

O *Spring*¹⁹ é uma *framework open source* para a plataforma *Java* que oferece diversas funcionalidades²⁰, das quais destaco as de maior importância para o contexto deste PEI:

- **Anotações para injeção de dependências:** as classes *Java* devem ser o mais modulares possíveis, para poderem ser usadas e testadas isoladamente. Isto é possível através da injeção de dependências que se baseia no princípio de inversão de controlo, onde a classe não se configura a si própria, mas é sim configurada por outra classe [25]. O *Spring* oferece funcionalidades para a gestão de objetos e os seus ciclos de vida, onde designa os seus objetos de *beans*. Estes são configurados através de um ficheiro XML, onde são guardadas as suas definições e dependências, no *Clinidata* o ficheiro tem o nome de “rpc-services-servelet”.

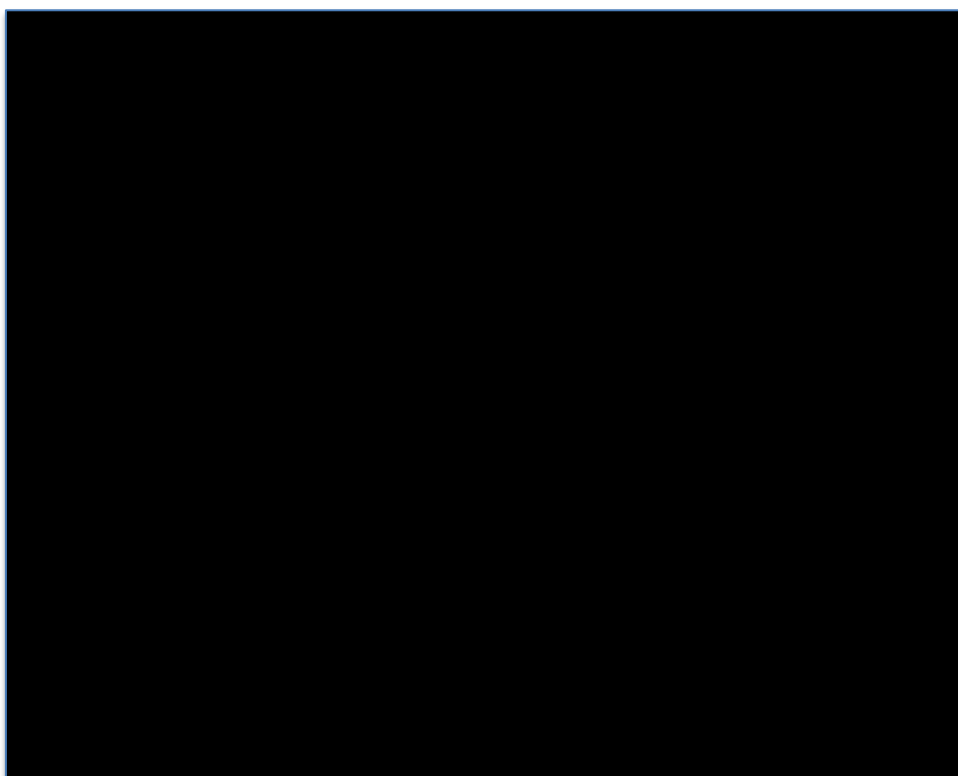


Figura 13. Exemplo de uso das anotações de Spring para injeções de dependências deste PEI

Na figura 13 temos um exemplo do uso das anotações de *Spring* onde o serviço²¹ criado para este PEI (ver secção 4.3.2) das doações (“DonationBC”) para

¹⁹ Spring, <http://www.springsource.org/>.

²⁰ Spring framework, <http://www.springsource.org/spring-framework>

²¹ Um Serviço (*service*) expõe um conjunto de métodos para serem invocados remotamente pela aplicação interna (por RPC) e/ou por aplicações externas (*WebServices*)

injetar outros serviços como o “ILabelBC” e também o DAO (*Data Access Object*) das doações (“IDonationDAO”). A anotação “@Autowired” marca um construtor “injetável” automaticamente através da injeção de dependências. Quando se corre a aplicação *Clinidata*, o *Spring* faz a correspondência entre as variáveis anotadas com “@Autowired” aos *beans* declarados no ficheiro “rpc-services-servelet” e as variáveis são injetadas após a criação do *bean* e antes de algum método poder ser invocado. A anotação “@Service” identifica a classe *Java* como sendo um serviço. Assim, quando é feito o arranque da aplicação do lado do servidor, é inicializado o *Spring*, que por sua vez instancia todos os *beans* que estão declarados no ficheiro XML e com a anotação “@Service”.

- **Suporte de transações:** uma transação numa BD corresponde ao conjunto de ações que são tratadas como um único bloco, deste modo, ou todas as ações são executadas ou nenhuma é. O *Spring* oferece um gestor de transações que propaga uma transação a vários métodos e fazer roll-back no caso de ter ocorrido uma exceção numa das ações do bloco.

A *framework Spring* cria uma camada de abstração que facilita o envio dos dados provenientes do front-end para o servidor com vista ao seu processamento. Esta camada de abstração é totalmente transparente para o Cliente, não existindo quaisquer custos pela sua utilização.

GWT

Google Web Toolkit ²² é uma *framework* SOFEA (*Service-Oriented Front-End Architecture* or *Service Oriented User Interface*), *open-source* e *web* para desenvolvimento rápido e *debugging* de *JavaScript* com uma aplicação com *front-end* implementado em *Java*, permitindo que:

- O código seja desenvolvido em *Java*
- Compilado em *JavaScript*
- Corrido no *browser*

Permite aos programadores criar aplicações *web* através de tecnologia *Ajax*, mas em linguagem de programação *Java*. O GWT suporta cliente-servidor, desenvolvimento e *debugging* em qualquer IDE orientado ao desenvolvimento em *Java*.

Esta ferramenta contém (i) um **GWT SDK** que fornece bibliotecas de API's de *Java* e de *Widgets*, tais como a *MaxFramework*, uma *framework* utilizada para componentes visuais da aplicação, permitindo implementar uma aplicação *AJAX*

²² GWT, <http://www.gwtproject.org/overview.html>

(Asynchronous JavaScript and XML) em *Java*, o seu objetivo é aumentar a produtividade do desenvolvimento, abstraindo do programador as diferentes especificidades entre *browsers* e *JavaScript* e tornar o seu desenvolvimento mais flexível, (ii) **GWT compiler** otimizado para correr todos os browsers, responsável pela conversão de código *Java* em *JavaScript* em fase de compilação e o *JavaScript* executado pelo *browser* para *Java*. [26]. Permite construir aplicações *web* complexas e otimizadas, retirando a responsabilidade por parte do programador de gerir os *XMLHttpRequest* e o código *JavaScript*.

Apesar de existir uma versão *GWT 2.8*, na *Maxdata* é utilizado o *GWT 2.6.1* uma vez que as versões mais recentes, para além de só permitirem o debug na *web* (a versão anterior permite *debug* em *web* e no *eclipse*) seria necessário realizar inúmeras alterações a vários componentes já desenvolvidos.

LiquidBase

LiquidBase [Radoslaw dziadosz 2017] é uma ferramenta *open-source* que possibilita o controlo de versão de BD através de ficheiros XML, YAML, JSON ou SQL. Estes ficheiros contêm *changeSets* que são identificados por ID e Autor. Ao correr o ficheiro de *LiquidBase*, as alterações à BD são efetuadas. É guardado um *changelog* na BD que identifica todas as alterações efetuadas e um *hash* da mesma. O *LiquidBase* suporta diferentes *SGBDs*

Na *Maxdata* esta ferramenta é utilizada para criar e alterar a estrutura da *BD*, nomeadamente a estrutura de diferentes tabelas, criação de *labels*, mensagens, opções, entre outras. Assim, todas as alterações à BD são colocadas num ficheiro XML designado *dbChangeLog.xml* (figura 14), que representa o estado que a BD deve ter, permitindo o controlo de versões da base de dados.

```
<changeSet author="raquel.alqueidao" id="229.387-66">
  <insert tableName="TBW_TEXT_LABEL">
    <column name="ID" valueNumeric="19131"/>
    <column name="LABEL" value="Assumir armazém da unidade de login"/>
    <column name="TEXT_GROUP_ID" valueNumeric="159"/>
  </insert>
</changeSet>
```

Figura 14 – Exemplo de um *changeSet* criado no ficheiro “*dbChangeLog*”

No contexto deste PEI, foram criados mais de 300 *changeSets*, incluindo as *labels*, tabelas, chaves primarias, chaves secundárias, mensagens para aparecerem nos ecrãs, entre outros.

GIT

O GIT é um sistema de controlo de versões distribuído adequado para projetos grandes, eficiente em recursos computacionais e com suporte para desenvolvimento não-linear (*branching*). Garante a integridade dos ficheiros, permitindo a programação simultânea por vários elementos da equipa de desenvolvimento. Sendo um sistema distribuído e devido ao uso de repositórios remotos, existe sempre backups caso algo corra mal. Como gere as versões do código fonte, torna possível reverter o código para uma versão anterior quando necessário.

O código do projeto *Clinidata* encontra-se no *branch master* do repositório GIT da Maxdata. Os *code-reviewers* são os únicos que podem fazer alterações no ramo *master*. Sempre que uma tarefa é atribuída, o programador tem que criar um *branch* novo partir do *master* com o número da tarefa a desenvolver. Cada tarefa vai ser registada no sistema de suporte da Maxdata, os seus tempos de desenvolvimento e o trabalho efetuado. Depois de terminada a tarefa, é necessário sincronizar o *branch* da tarefa com o *master* e fazer um *merge-request*²³ a um dos *code reviewers*. Se na *review* não foram necessárias alterações ao código desenvolvido, segue para testes para a equipa de assistência, caso contrário volta para traz, onde o programador em questão corrige os pontos da *review* e o processo continua até não existir pontos para corrigir.

GitLab

GITLAB é um repositório online que tem como principal funcionalidade a gestão de repositório GIT, assim como a criação de *wikis* e um *issue tracker*, existindo um por projeto. A Maxdata utiliza a versão *open-source* distribuída sobre a licença MIT, para ser utilizado para a submissão de *Merge Requests*.

No contexto do meu PEI, sempre que terminava uma tarefa, realizava um *merge request* para o *branch* “master banco de sangue”, onde ficava a aguardar o *feedback* do *code reviewer*, se não houvesse correções reportadas a fazer, o *branch* é *merged* no *branch* “master banco de sangue”, caso contrário, era necessário a sua alteração até ir de acordo com o exigido.

4.2 Fluxo de sequência

Após a análise dos requisitos funcionais descritos no documento de análise, podemos definir o fluxo deste módulo de dados como o mostrado na figura 15.

²³ Pedido para o *branch* da tarefa ser integrado no *master* do Clinidata

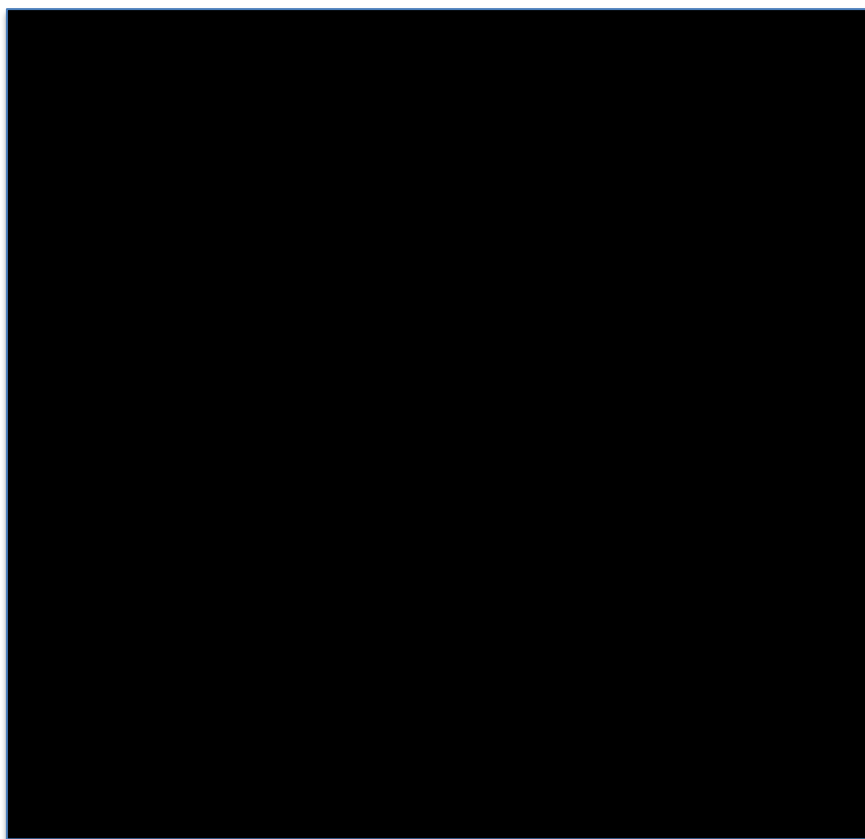


Figura 15. Diagrama de fluxo do módulo gestão de dadores (Elaboração própria)

O processo de doação é iniciado pela **consulta**, onde é realizado uma pré-triagem, se esta for aprovada, o dador segue para colheita, caso contrário é dispensado. Antes de iniciar **colheita** é realizado uma **medição** vital ao dador para saber se está com os valores dentro do esperado, se sim, inicia-se a colheita, senão é registado o motivo e o dador é dispensado. Após a colheita feita é realizada outra medição final e a dádiva é colocada num *kit* e prossegue para o laboratório para ser **testada e analisada**, quando o seu tipo de sangue estiver determinado e se não falhou nos testes, a unidade segue para decomposição de unidades, caso contrário é imediatamente destruída(**sangria**). Na **decomposição** de unidades é onde se define os componentes a obter a partir da dádiva, saindo etiquetas a indicar os diferentes componentes que se quer obter. Posteriormente é feito a **validação** ou rejeição das unidades para serem transfundidas, essa decisão é feita quando os resultados dos exames às componentes estiverem prontos. Se os resultados forem negativos, é destruída (sangria), senão são **disponibilizadas para stock** para serem usadas por uma ou mais transfusões.

4.3 Arquitetura do Sistema

A aplicação encontra-se dividida em três camadas, camada de dados (4.2.1) onde assenta o modelo de dados, a camada de negócio (4.2.2) onde se encontra a lógica

inerente ao negócio e a camada cliente (4.2.3) onde é desenvolvida a interface disponibilizada ao utilizador. É possível observar a interação entre as diferentes camadas aplicacionais na Figura 16.

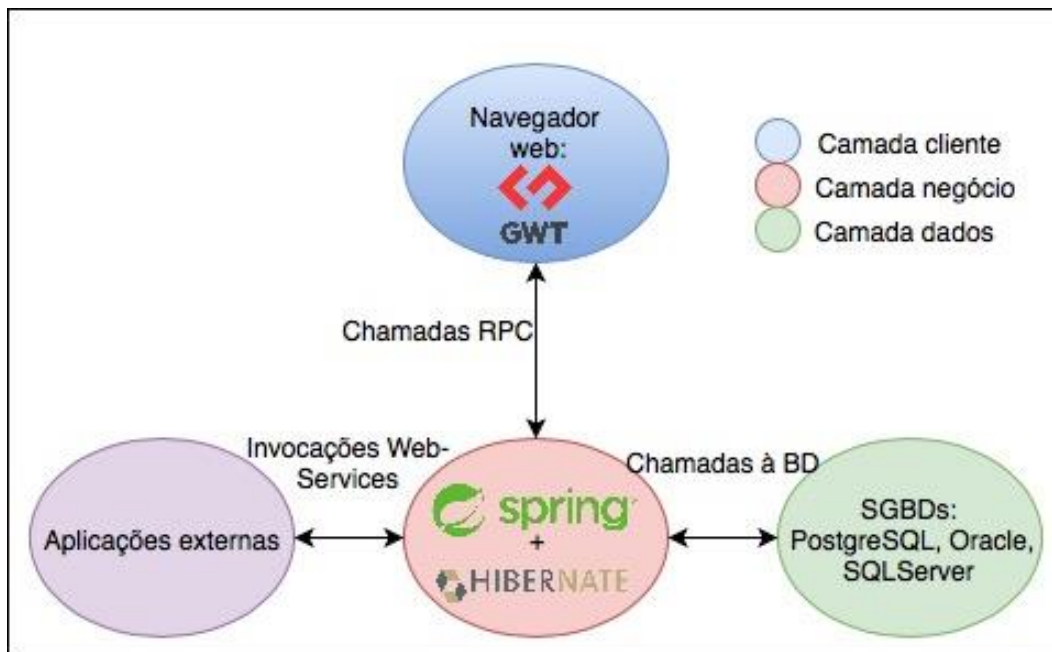


Figura 16. Diagrama de interações entre as camadas aplicacionais (Elaboração própria)

No projeto da *Clinidata* existe

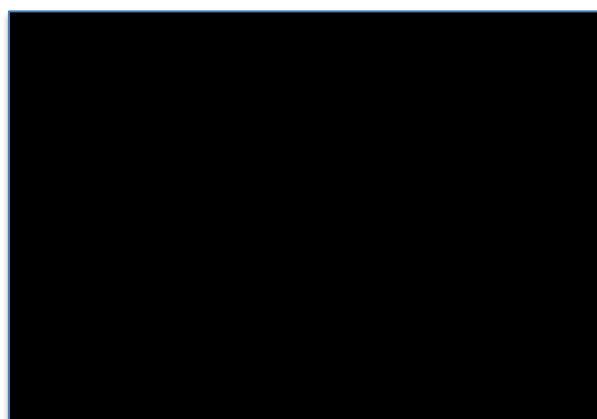


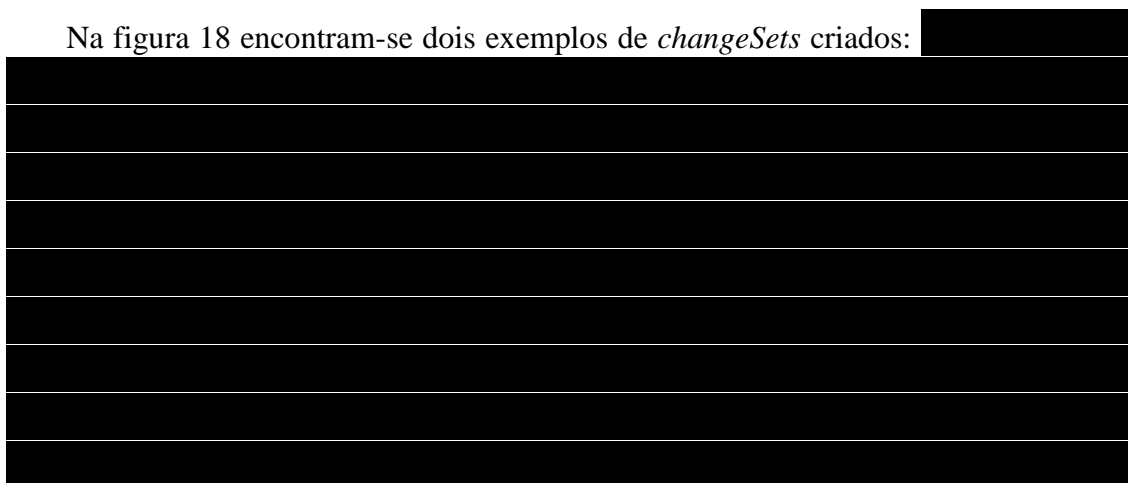
Figura 17. Vista geral dos *packages* do *Clinidata*

4.3.1 Camadas de dados

O modelo de dados da aplicação é um modelo relacional e os dados poderão ser armazenados em diferentes SGBDRs (Sistemas de Gestão de Base de Dados Relacionais), nomeadamente *Oracle Server*, *Derby* e *PostgreSQL*. Permite uma representação lógica e consistente da informação inerente à aplicação. Neste PEI, os sistemas de gestão de base de dados utilizados foram o *Oracle Database*²⁴ e *PostgreSQL Server*²⁵. A maioria dos clientes da Maxdata utiliza o primeiro tipo de SGBD, apesar de terem semelhanças em diversos aspetos, existem algumas diferenças a nível de sintaxe do *SQL* [27] e a base de dados *Oracle* fornece um melhor desempenho [28]. No entanto as licenças para o Oracle são dispendiosas, levando a que os clientes de menor dimensão de dados optem por sistemas como o *PostgreSQL*.

No contexto deste PEI fui responsável pela criação de todas as tabelas inerentes ao módulo de dados. Para cada entidade detalhada na secção 3.2.1, foi criada a sua tabela e modelo de domínio, para tal foi necessário a criação de vários *changeSets* para a criação das tabelas, com a sua estrutura, chaves primárias, chaves secundárias, índices e sequências para a criação de ids únicos, que foram adicionados ao ficheiro “dbChangeLog”. Este ficheiro representa todas as tabelas, estruturas e relações do *sw Clinidata*. Ao correr-se o *LiquidBase*, vai correr o ficheiro *XML*, ele consegue detetar que *changeSets* já foram corridos ou não, e atualiza a BD com os novos domínios desejados. A BD utilizada durante o PEI para a criação das tabelas e o seu preenchimento, foi realizada numa BD de *PostgreSQL*, num *schema-dev* utilizado pela equipa de I&D, pelo que não lidei com nenhuns dados vindos do cliente. pelo que não lidei com nenhuns dados vindos do cliente.

Na figura 18 encontram-se dois exemplos de *changeSets* criados:



²⁴ Oracle Database, <http://www.oracle.com/technetwork/database/enterpriseedition/overview/index.html>

²⁵ PostgreSQL, [<http://www.postgresql.org/>]



Figura 18. ChangeSet para a criação de uma tabela e a sua chave

Para além da criação dos changeSets das tabelas foi necessário criar os modelos de domínios que vão representar os diferentes objetos das tabelas descritas anteriormente (exemplo criado na figura 19). Usou-se a anotação *@Entity*, como se pode verificar na figura seguinte. O nome colocado em *@Table* corresponde ao nome dado no *changeSet* quando a tabela foi criada.

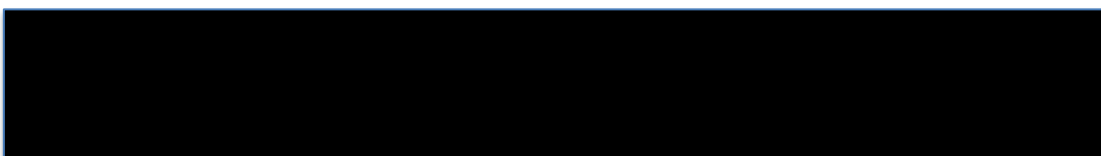
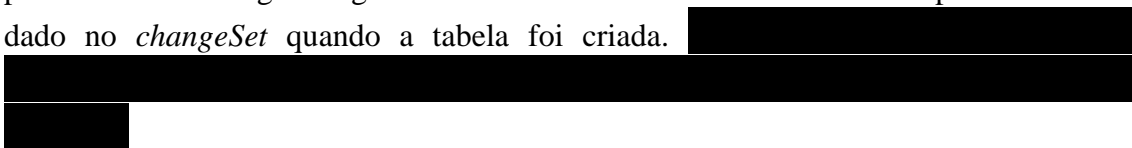


Figura 19. Modelo de domínio criado para a tabela de doação “DBB_DONATION”

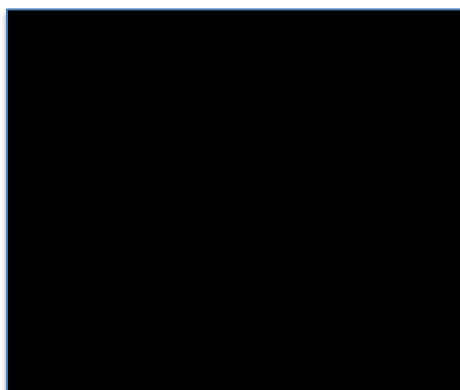


Figura 20. Versão geral do package “shared/bloodBank”

4.3.2 Camada de negócio

Nesta camada encontra-se toda a lógica de negócio, responsável pelos processos e rotinas que realizam operações sobre os modelos de domínio. Como foi possível observar na figura 16, esta camada é o ponto de entrada do servidor, a camada cliente nunca comunica com a camada de dados diretamente. O cliente pode inclusivamente ser uma aplicação externa que pretende comunicar com a nossa aplicação (círculo roxo Figura 16). Nas aplicações na área da saúde é bastante comum existirem integrações com aplicações externas para que terceiros os possam consumir, como a exportação de dados, o contrário também é válido, a aplicação também atua como fornecedor de informação, essas integrações são implementadas nesta camada. Assim disponibilizamos *services*²⁶ para que os seus métodos expostos sejam consumidos por aplicações internas (RPC) e externas (*Web-Services*). As integrações *Web-Services* não estavam no âmbito do meu *PEI*, tendo sido desenvolvidas por um programador sénior.

No contexto deste *PEI*, fui responsável pela implementação na camada de negócio, do *service* e do *repository* do módulo de dadores. Para tal foi necessário:

Service das doações (“**DonationBC**”)

1.

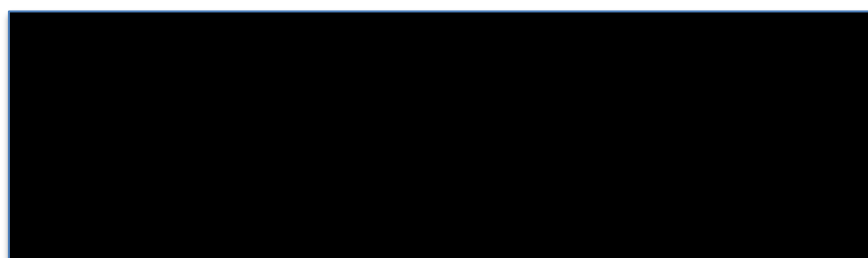
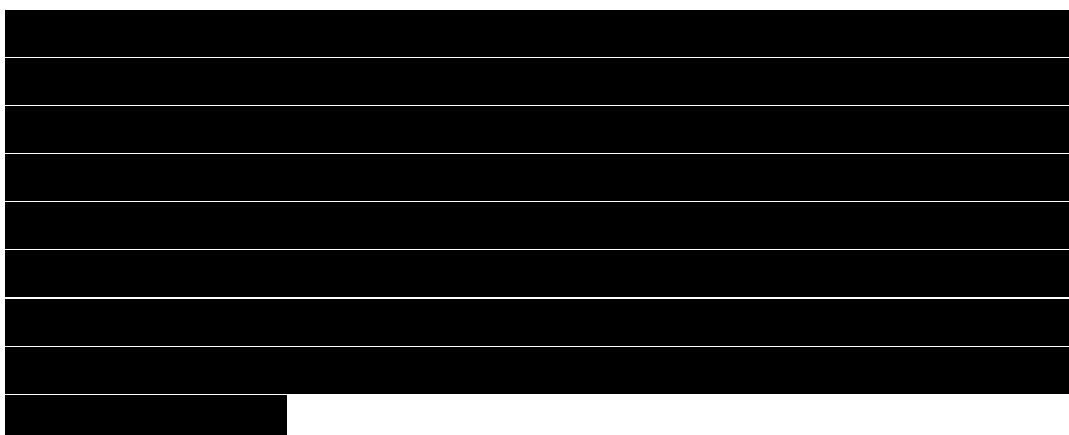
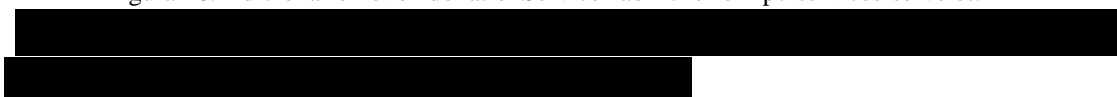


Figura 26. Adicionar o novo “donationService” ao ficheiro “rpc-services-servelet.xml”



2.



Figura 27. Classes “IDonationBC” e “IDonationBCAsync”

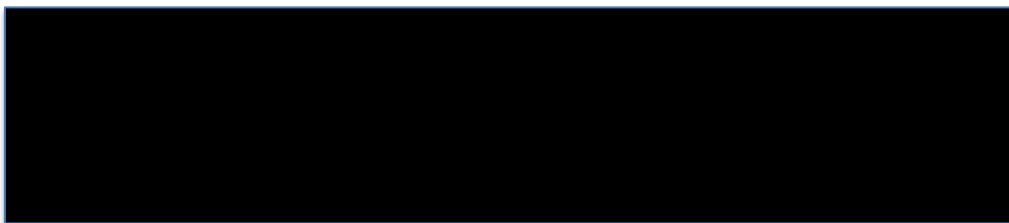


Figura 28. Classe “DonationBC” (*service* das doações)

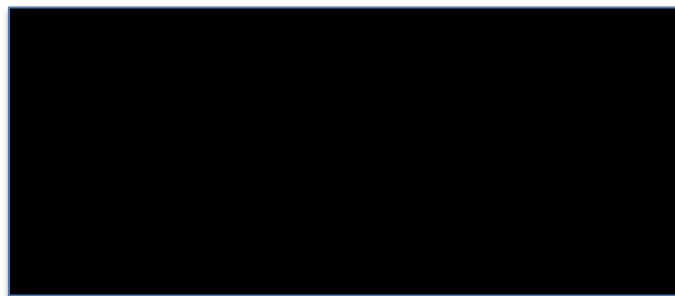


Figura 29. Exemplo de método criado no “DonationBC” (*service* das doações)

Repository das doações (“DonationDAO”)

1.



Figura 30. Interface “IDonationDAO”(interface do repository das doações)

2.

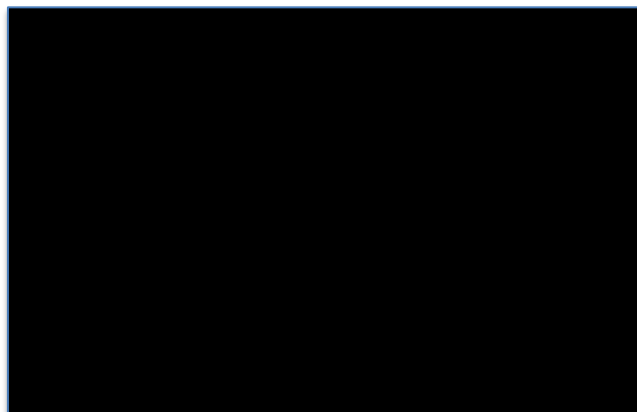


Figura 31 Classe “DonationDAO”(repository das doações)





Figura 32 Visão geral do package “server”



Figura 33 Visão geral do package “shared”

4.3.3 Camada Cliente

A utilização do GWT veio facilitar a implementação da camada cliente deste projeto como foi referenciado na seção 4.1, permitindo ao programador implementar a camada de negócio e cliente na linguagem em Java permitindo troca de código entre as diferentes camadas. Ficou da responsabilidade do GWT, a conversão de *Java* para *JavaScript* e HTML para o *browser*, e vice-versa para receber na camada de negócio pedidos da camada cliente por RPC ou por um serviço externo via *web-services*.

A definição de um padrão de desenho é importante para permitir uma maior organização no desenvolvimento de código, evidenciando as diferentes funcionalidades, permitindo uma melhor manutenção do código no futuro. De entre os diversos padrões de desenho populares a nível arquitetural [29], como o MVC (*Model-View-Controller*) concluiu-se que o MVP (*Model-View-Presenter*) é a arquitetura que se melhor adequa ao desenvolvimento de aplicações *GWT*²⁷, removendo o máximo de lógica da *view* para conseguir construir e testar a interface gráfica (GUI – *Graphical User Interface*). Como se pode observar na figura 34, no modelo MVC, quando a *view* recebe um evento do utilizador, altera o *model* através de um *controller*, mas também pode aceder diretamente ao *model* para alterar o seu estado ou para mostrar *outputs* ao utilizador. No MVP, a *view* nunca acede diretamente ao *model* e o *presenter* assume um papel de mediador entre a *view* e os objetos do modelo de domínio, comunicando à *view* para alterar o seu estado em resposta a *inputs* do utilizador ou a mudanças do modelo. Este padrão é muito semelhante ao MVC, contudo, invés de a lógica de apresentação ser

²⁷ Google Web Toolkit, <http://www.gwtproject.org/>.

partilhada entre o *controller* e a *view*, no MVP esta lógica encontra-se no *presenter*, [30].

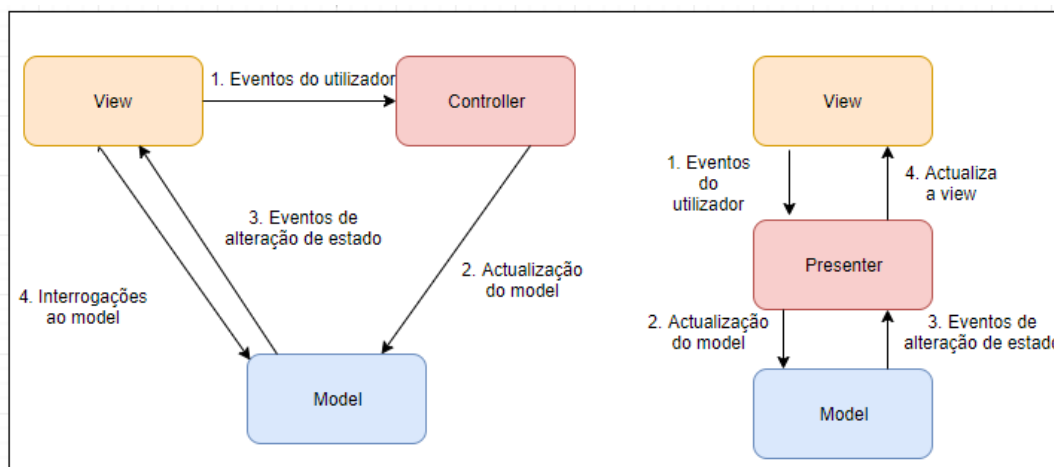


Figura 34. Padrão arquitetural MVC vs. MVP

Todo o software do *Clinidata* segue um modelo MVP, onde o **model** representa o modelo de dados do domínio, os objetos a serem manipulados e define a interface dos dados a serem manipulados/exibidos. A **view** define a interface gráfica onde são exibidos os dados do *model* e dispara eventos mediante comandos/operações executadas pelo utilizador. Por último, o **presenter** atua sobre o *model* e a *view*, onde através de repositórios obtém os dados do *model* e injeta-os na *view*; é no *presenter* que se situa a lógica inerente ao domínio da aplicação. As interações do user na *view* desencadeiam ações a serem executadas no *presenter*.

4.3.4 Packages criados no *Clinidata® Transfusion*

Na figura 25 encontra-se os diferentes packages deste módulo *Clinidata®Transfusion* (“*client/bloodBank*” e), os que se encontram dentro de um retângulo vermelho correspondem aos implementados por mim durante este PEI, todos eles seguindo o modelo MVP e cada um representa um ecrã a exibir.

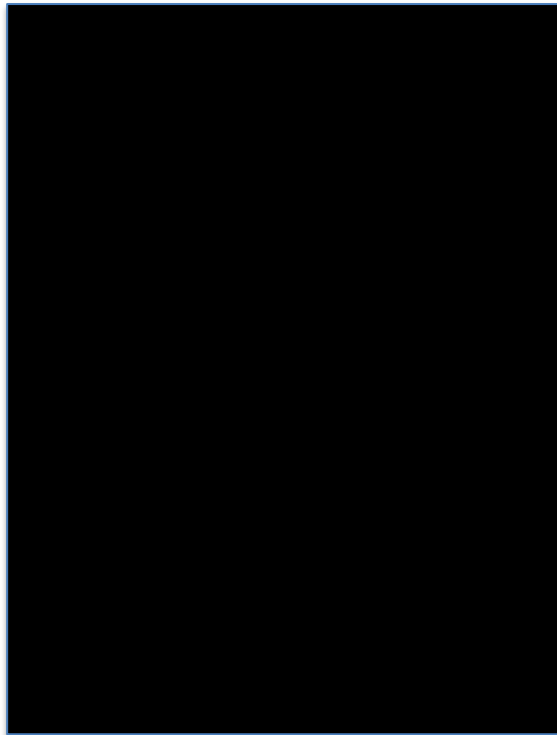


Figura 35. Visão geral do *package* “cliente” do banco de sangue

No contexto deste PEI, para além dos novos *packages* criados, também foi necessárias alterações a módulos existentes de modo a funcionarem coerentemente com o novo módulo, como é o caso da geradora do banco de sangue, tendo sido adicionado novas funcionalidades.

4.4 Casos de uso

Nesta seção são apresentadas algumas particularidades do *Clinidata* e casos de uso dos ecrãs implementados no contexto deste PEI. Para cada ecrã apresentado, foi criado um *package* no *clinidata* que segue o modelo MVP.

4.4.1 Configuração de ecrã

Este caso de uso serve para contextualizar a aplicação *Clinidata*, da qual o módulo implementado neste PEI se integra. É apresentado o ecrã de “login” (Figura 37), o aspeto do ambiente de trabalho e um exemplo de uma pesquisa no menu da aplicação (Figura 38). Este caso de uso não foi implementado por mim nem é do contexto deste PEI.

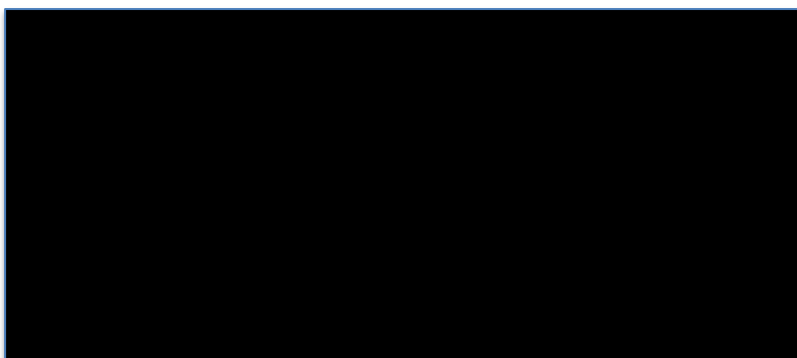


Figura 37. Ecrã de “login” da aplicação Clinidata

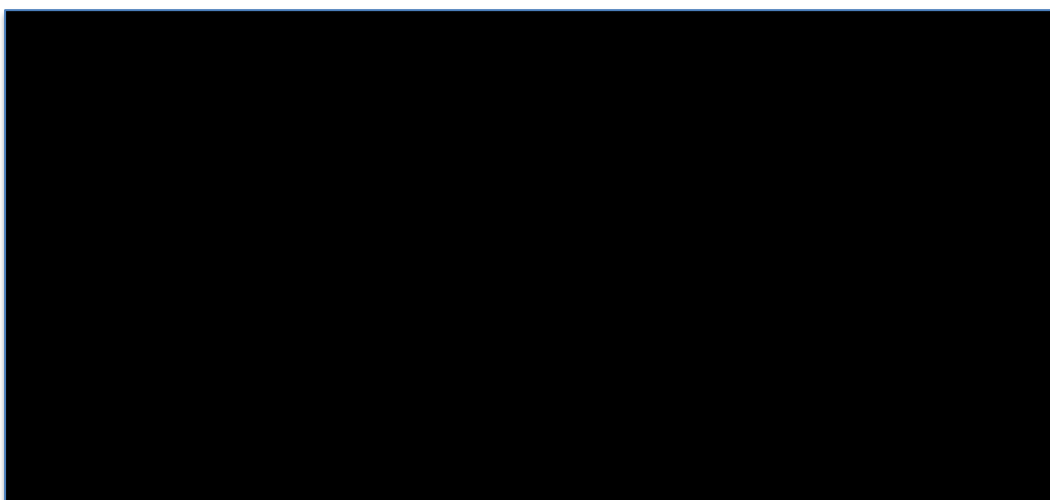


Figura 38. Ambiente de trabalho do utilizador na aplicação Clinidata (esq) e exemplo da pesquisa de um ecrã de exibição (direita)

O *Clinidata*® contém dois conceitos importantes sobre os seus ecrãs:

1. **Ecrãs de configuração** (geradora²⁸):



²⁸ Expressão utilizada na *Maxdata* para referenciar o ecrã de configuração

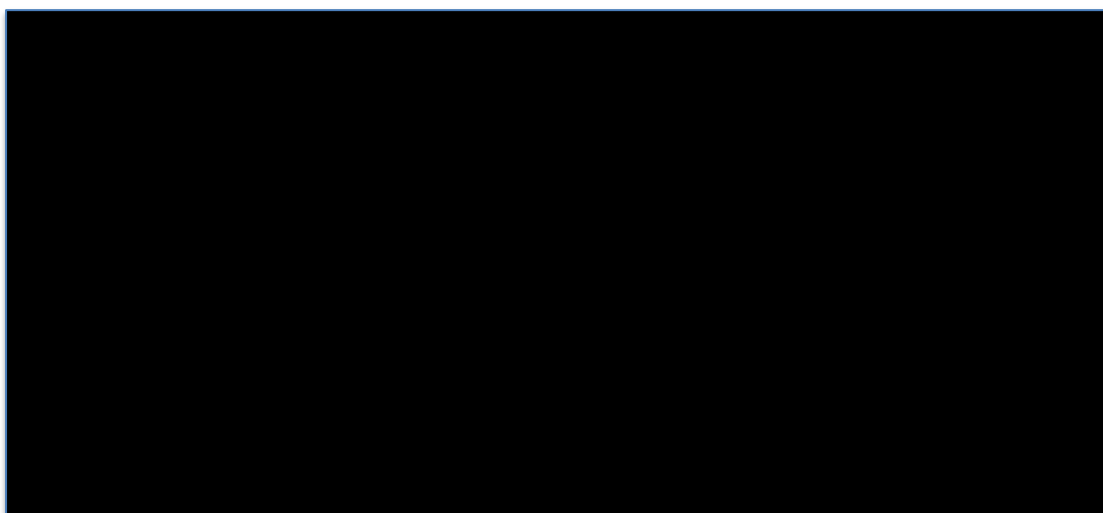
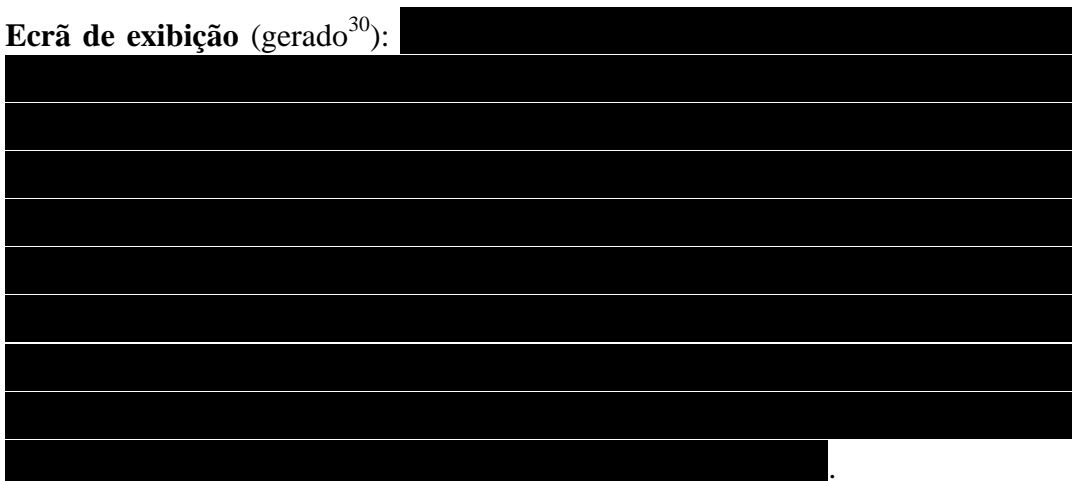


Figura 39. Exemplo de configuração da geradora do banco de sangue para exibir o ecrã de “Gestão de doações”

2. **Ecrã de exibição** (gerado³⁰):



²⁹ Representa o texto que queremos usar para identificar um campo. São adicionados ao ficheiro dbChangeLog.xml a inserção na tabela TBW_TEXT_LABEL com o id da *label* e uma *string* com o texto a aparecer.

³⁰ Expressão utilizada na *Maxdata* para referenciar um ecrã de exibição

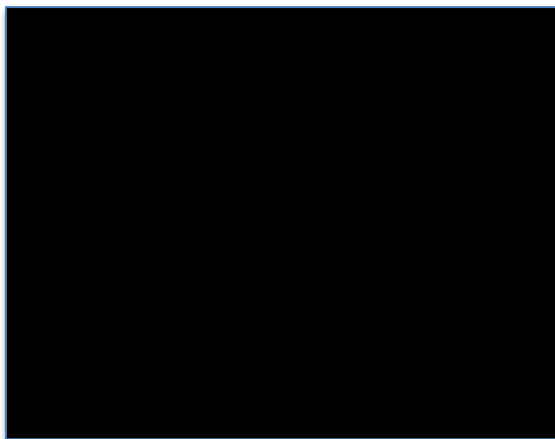


Figura 40. Visualização do menu inicial do Clinidata

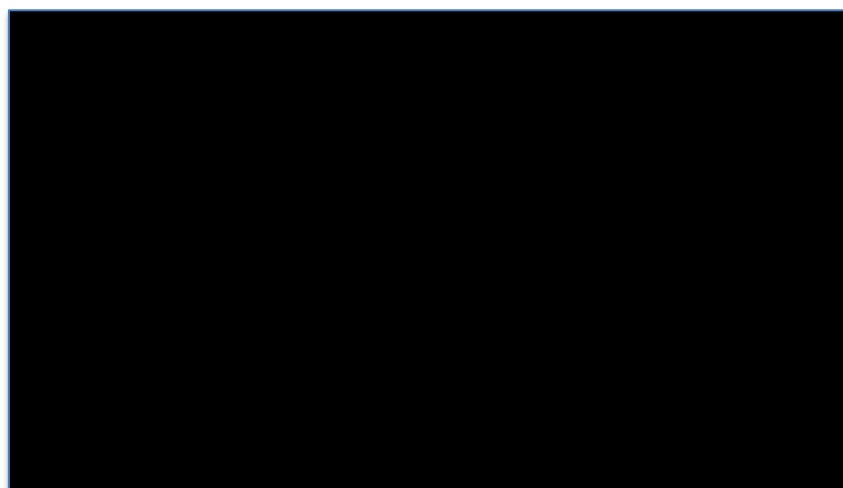
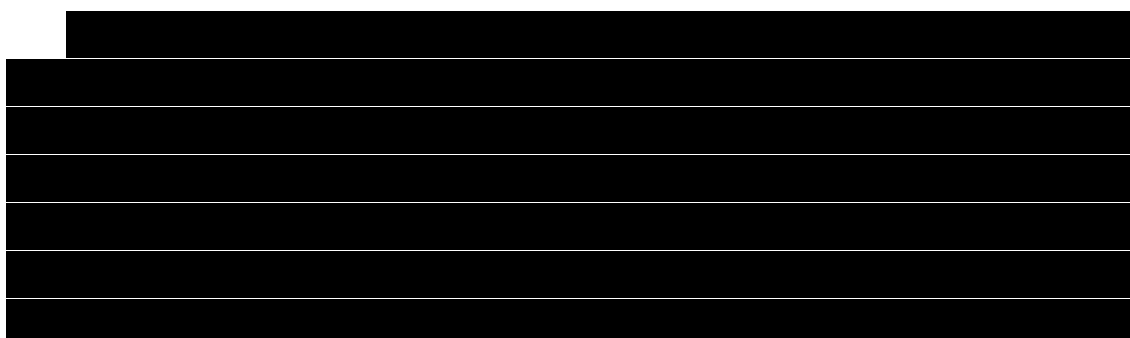


Figura 41. Exibição do ecrã “Gestão de doações” com id 39655

4.4.2 Gestão de doações



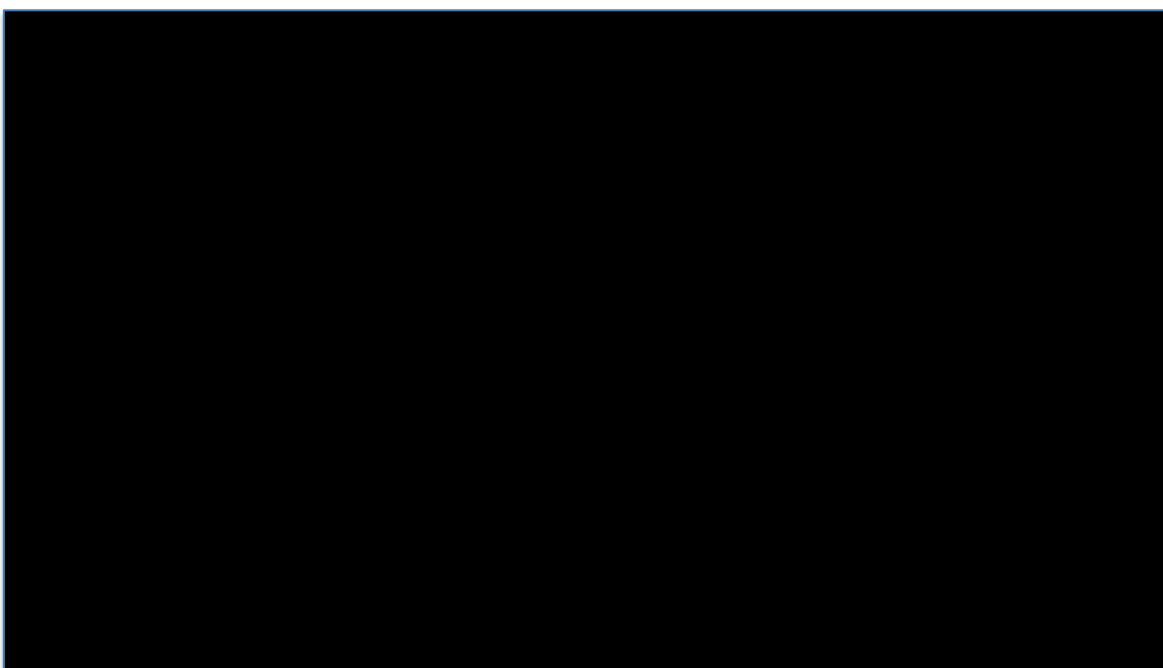
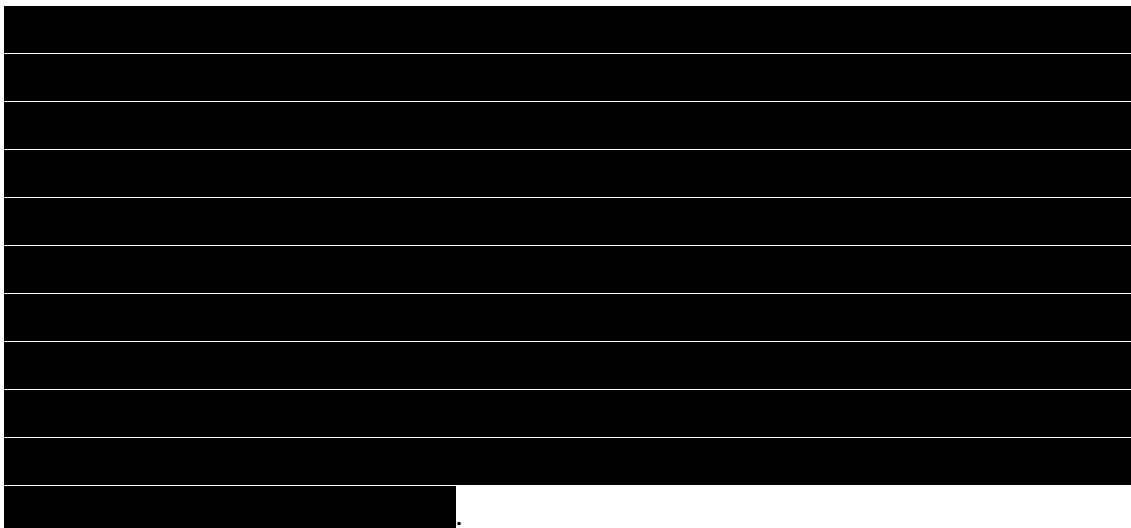


Figura 42. Ecrã “Pesquisa” (cima drt) e *grid* de resultados(baixo) da Gestão de doações

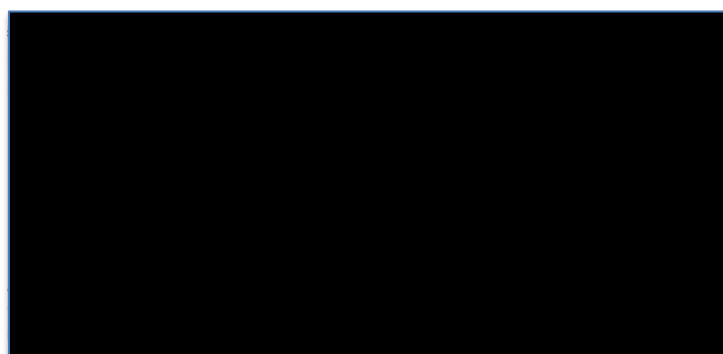



Figura 43. Ecrã aberto depois de ter clicado em “Imprimir” na fig.42 (esq) e Pop-up de impressão após ter clicado em “Imprimir” (drt)

Este módulo de impressão já existia no projeto do *Clinidata*, foi necessário adicionar os campos do âmbito das doações ao código já desenvolvido, 



4.4.3 Detalhe de doação

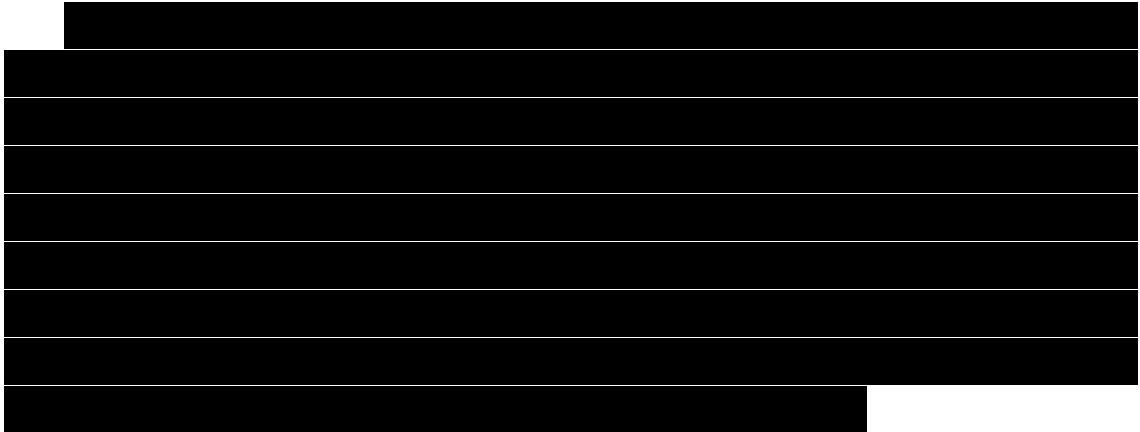
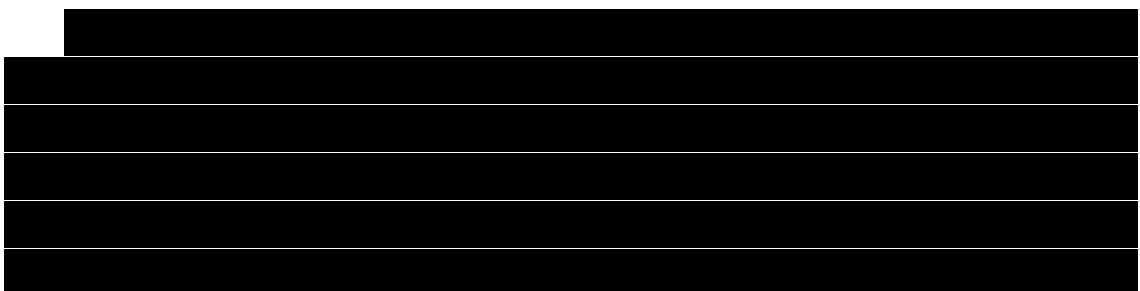


Figura 44. Ecrã “detalhe de doação”

4.4.4 Decomposição de unidades



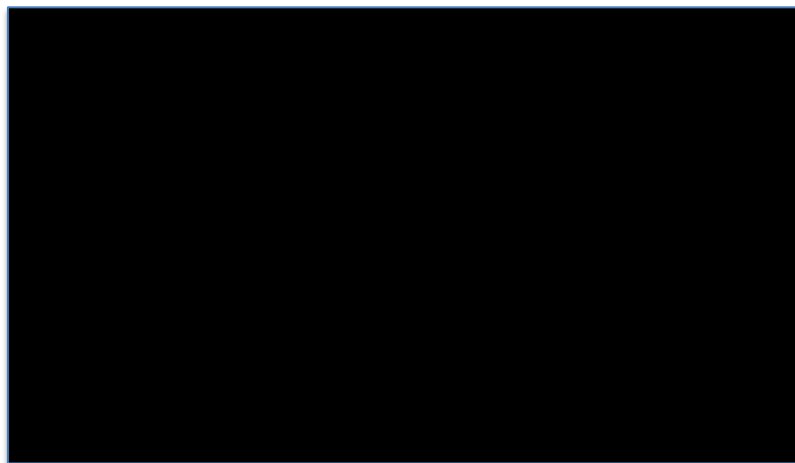
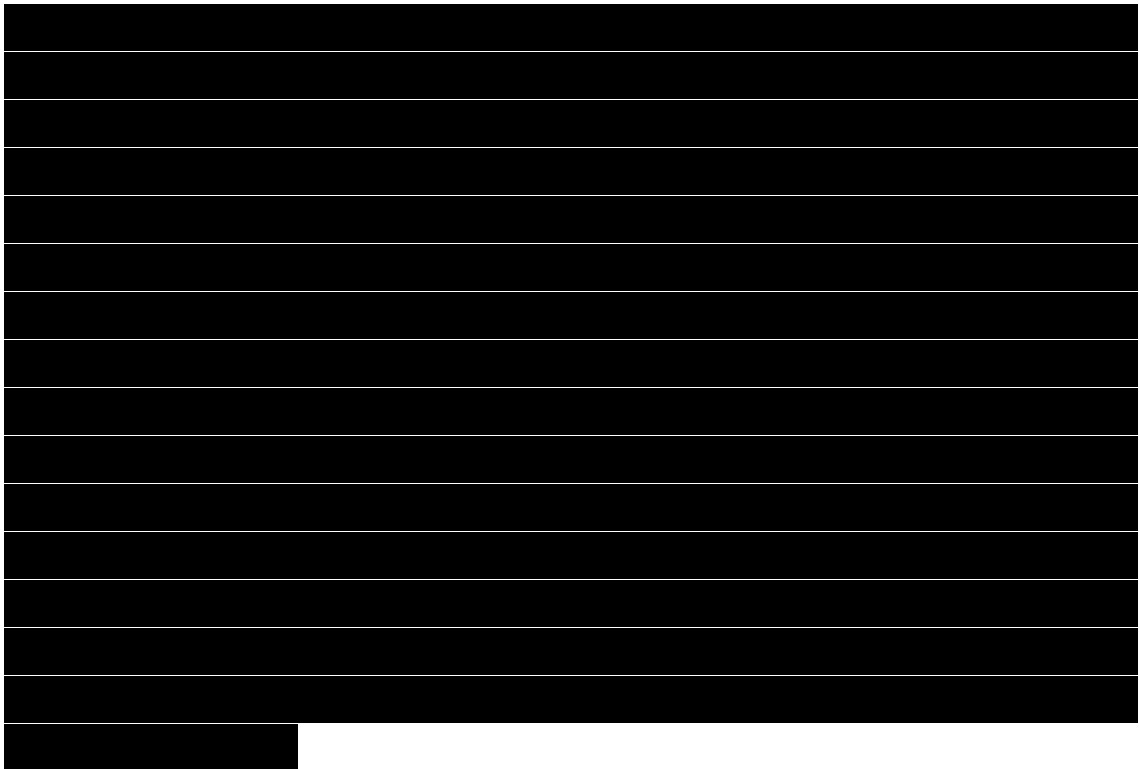


Figura 45. Ecrã “Decomposição de unidades”

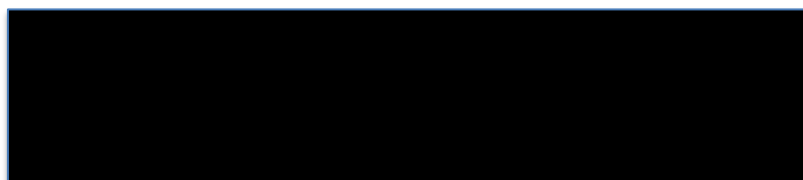


Figura 46. Resultado após ter criado unidade de Sangue Total

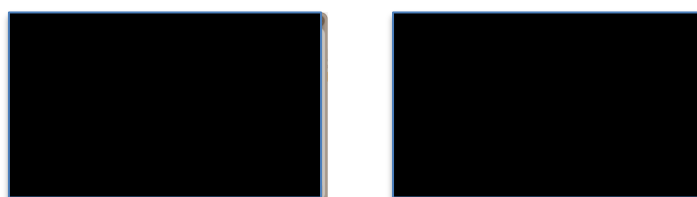


Figura 47. Ecrã “Inserir unidade” (esquerda) e Ecrã “Editar unidades” (direita)

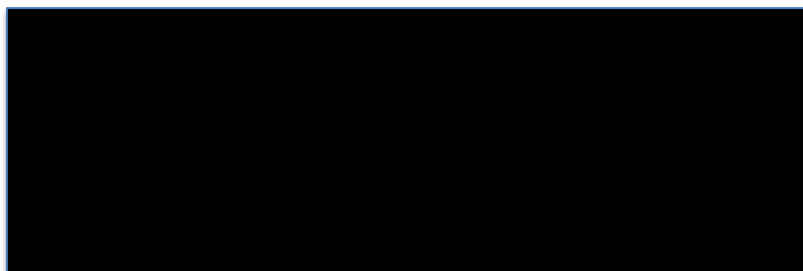


Figura 48. Ecrã de impressão de etiquetas “iniciais” das unidades



1 → [Barcode: A9999 12 123456 8] ← 2

3 → [Data da coleta: 05 JAN 2011] Rh POSITIVO

4 → [Barcode: 61001100] Doador Voluntário

5 → [Data de validade/Tempo: 31 JAN 2012 15:20]

6 → [Barcode: 61001100]

CONCENTRADO DE HEMÁCIAS
Adenine-Salino (AS-1) Adicionada
De 450 ml CPD Sangue Total
Armazenar entre 1 e 6 C

Negativo para anticorpos para CMV

- 1 Número de Identificação da Doação
- 2 ABO/Rh Grupo Sanguíneo
- 3 Data da Coleta (opcional)
- 4 Código do Produto
- 5 Data da Validade (e hora)
- 6 Testes Especiais(opcional)

Figura. 49 Etiqueta para a unidade segundo a norma IBST128 (adaptado de ³¹)

4.4.5 Medições vitais



³¹ <http://www.iccbba.org/uploads/77/b5/77b5b71bc50f4634607797b03b22d78b/IN-028-An-Introduction-to-IBST-128-Portuguese-v1.pdf>

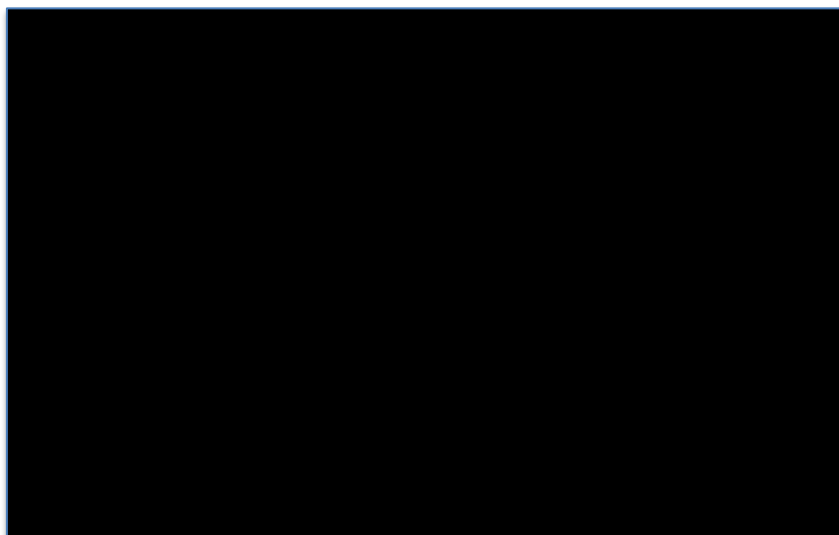
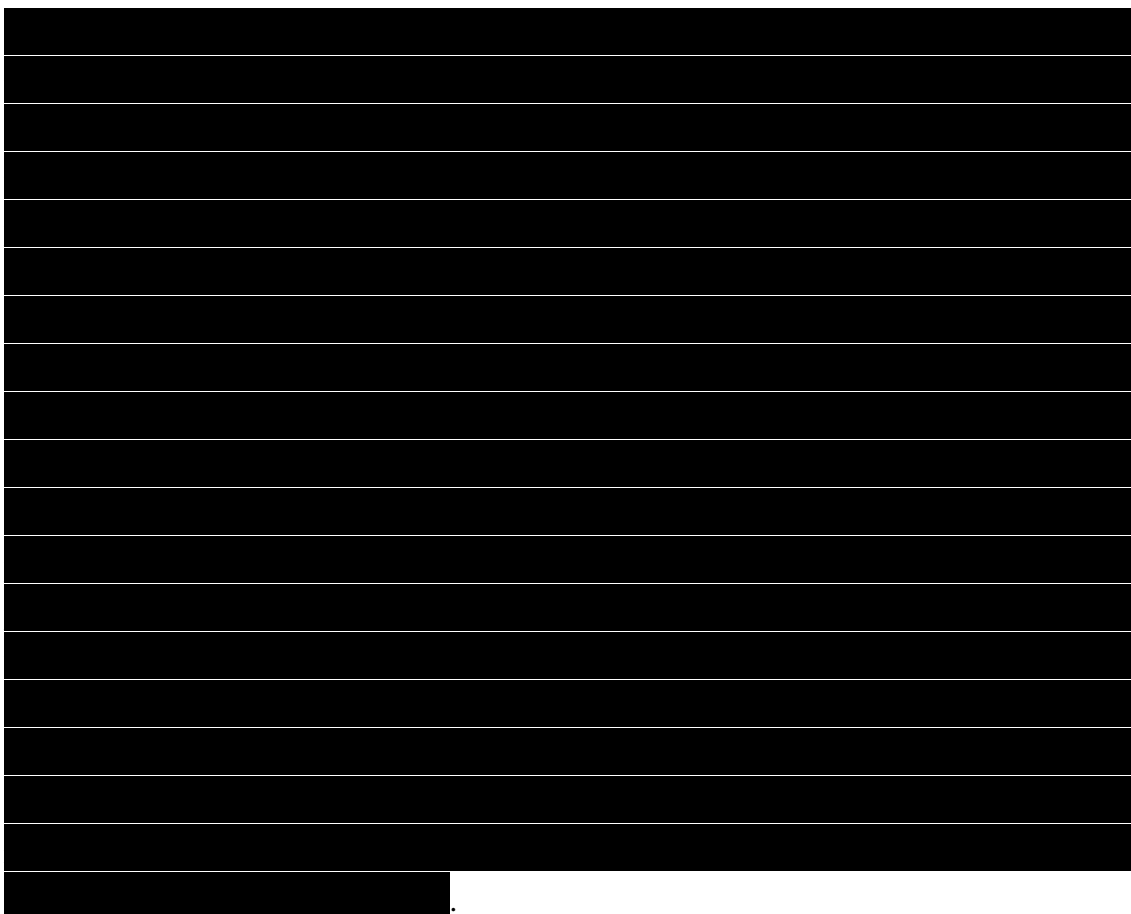


Figura 50. Ecrã de “Medições vitais” (âmbito doações)

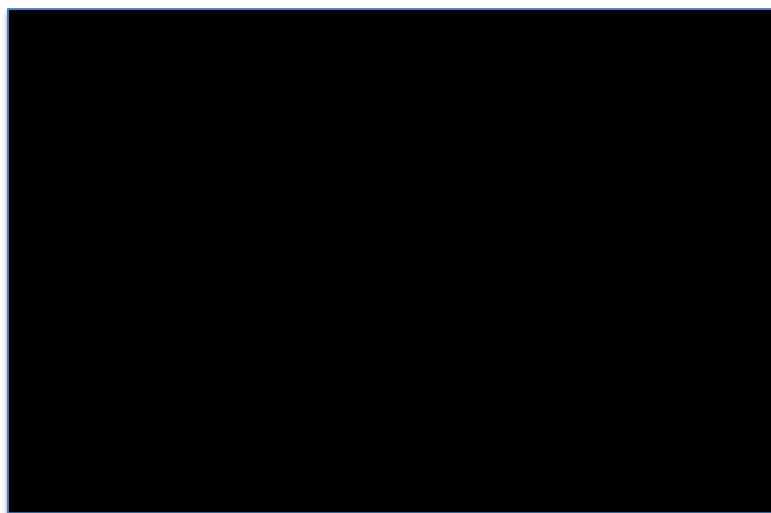


Figura 51. Ecrã de “Medições vitais” quando é clicado em “alterar” com uma medição selecionada vital (âmbito doações)

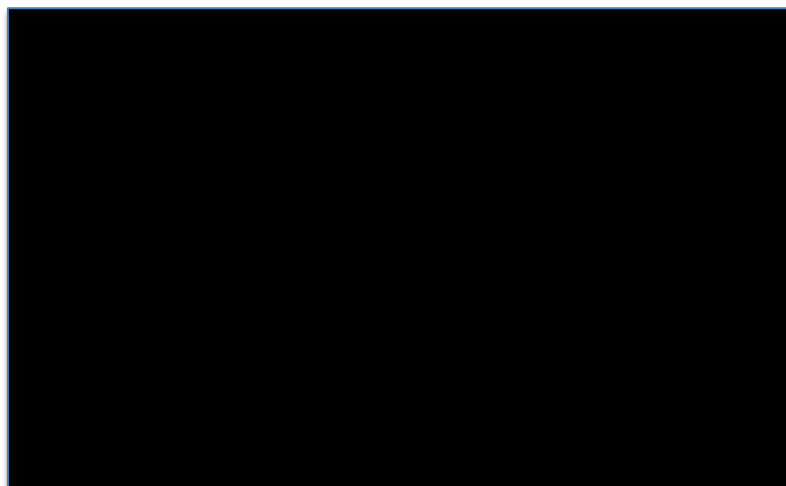
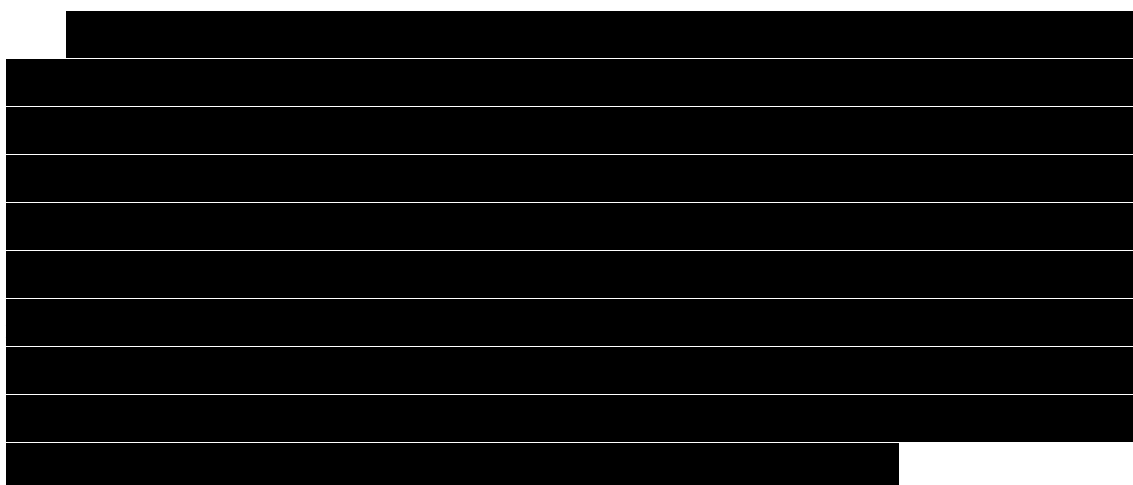


Figura 52. Ecrã de “Medições vitais” (âmbito transfusões)

4.4.6 Validação de unidades



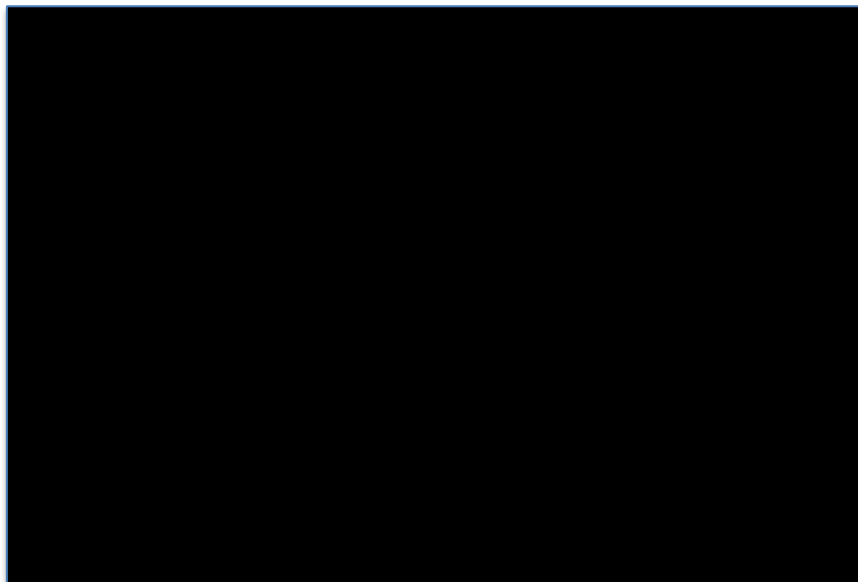
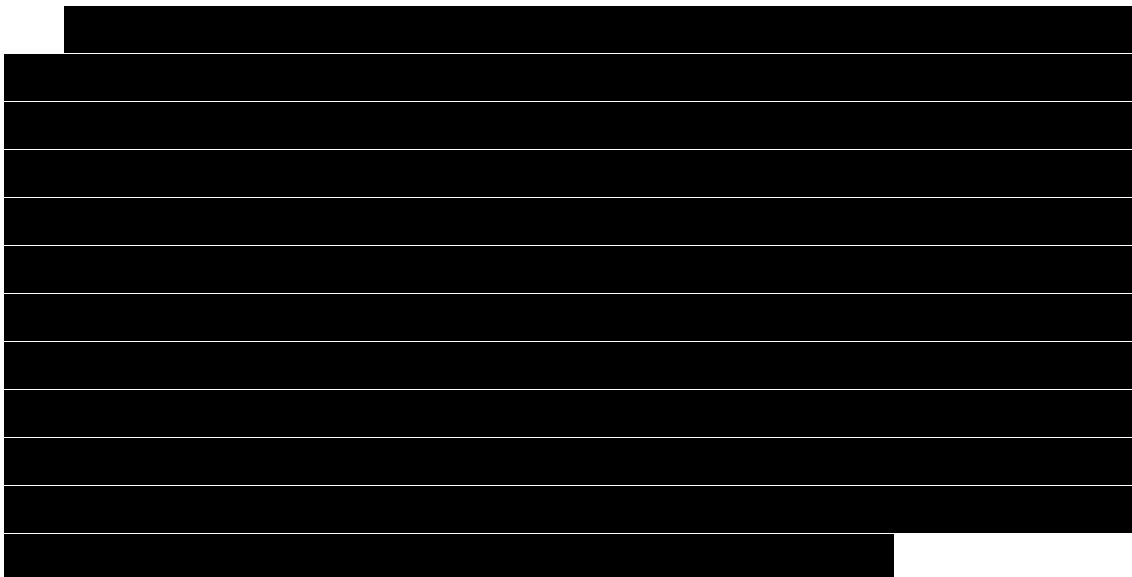


Figura 53. Ecrã de “Validação de unidades”

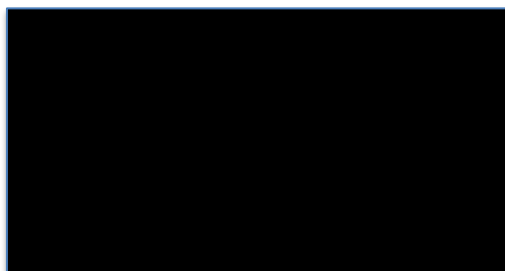


Figura 54. Ecrã “Alterar unidade”

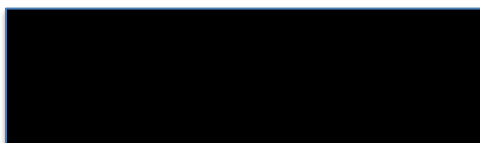


Figura. 55 Ecrã de impressão das etiquetas “finais” da unidade

4.4.7 Gestão de transfusões

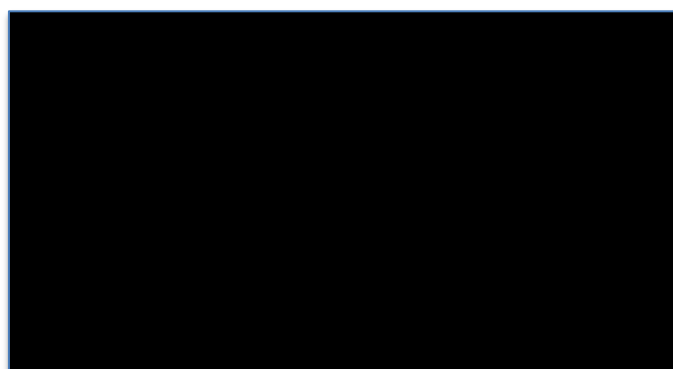
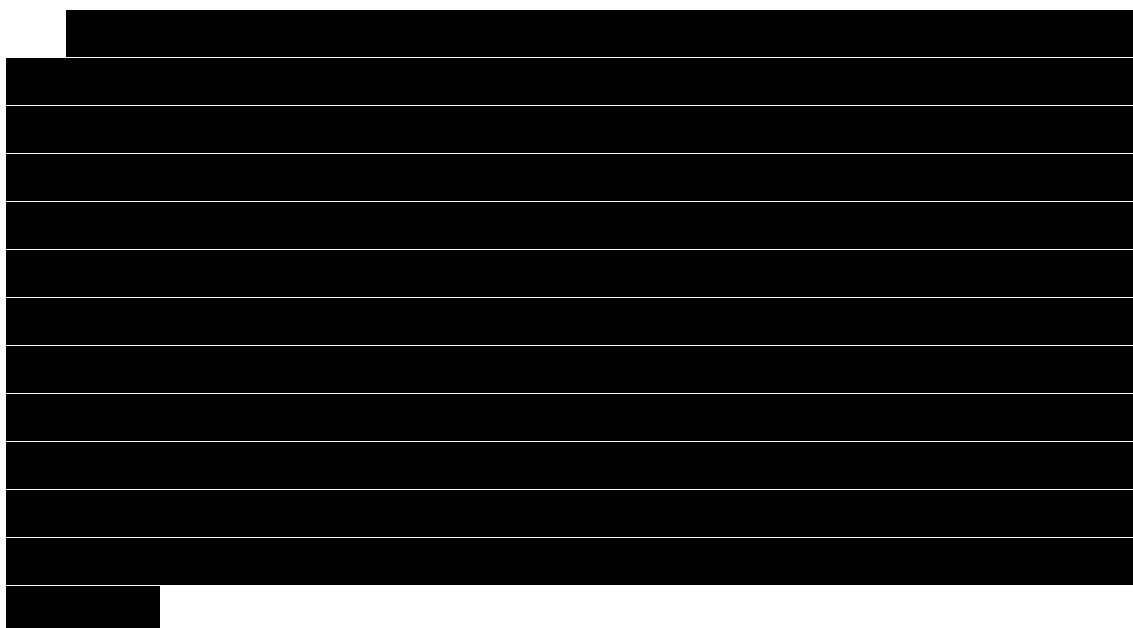


Figura 56. Ecrã de “Gestão de transfusões” (Pesquisa de transfusões)

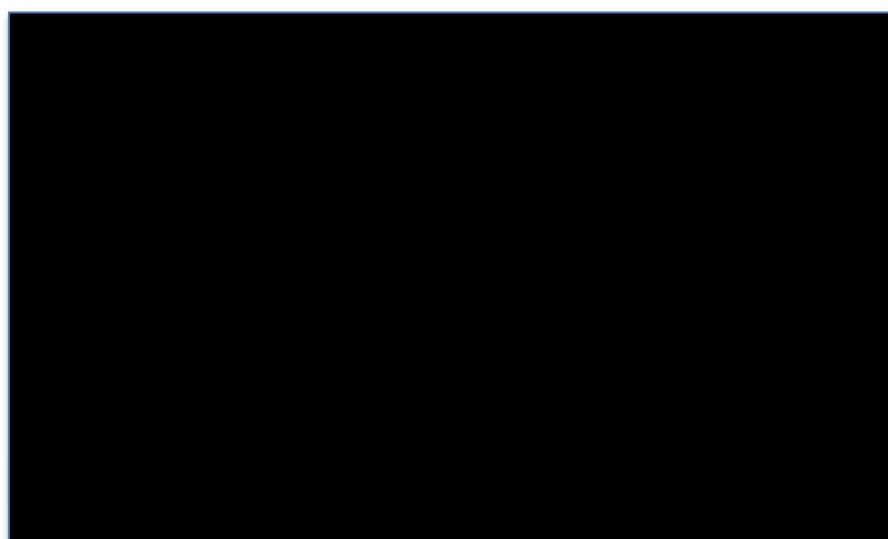


Figura 57. Ecrã de “Gestão de transfusões “(*Grid* de resultados)

4.4.8 Detalhe de transfusão

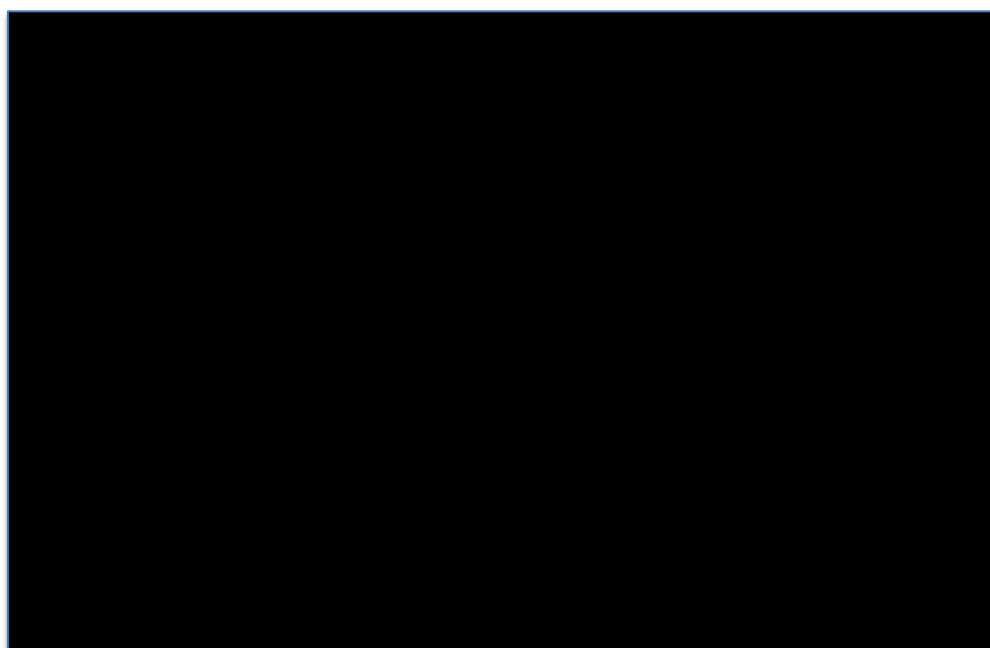
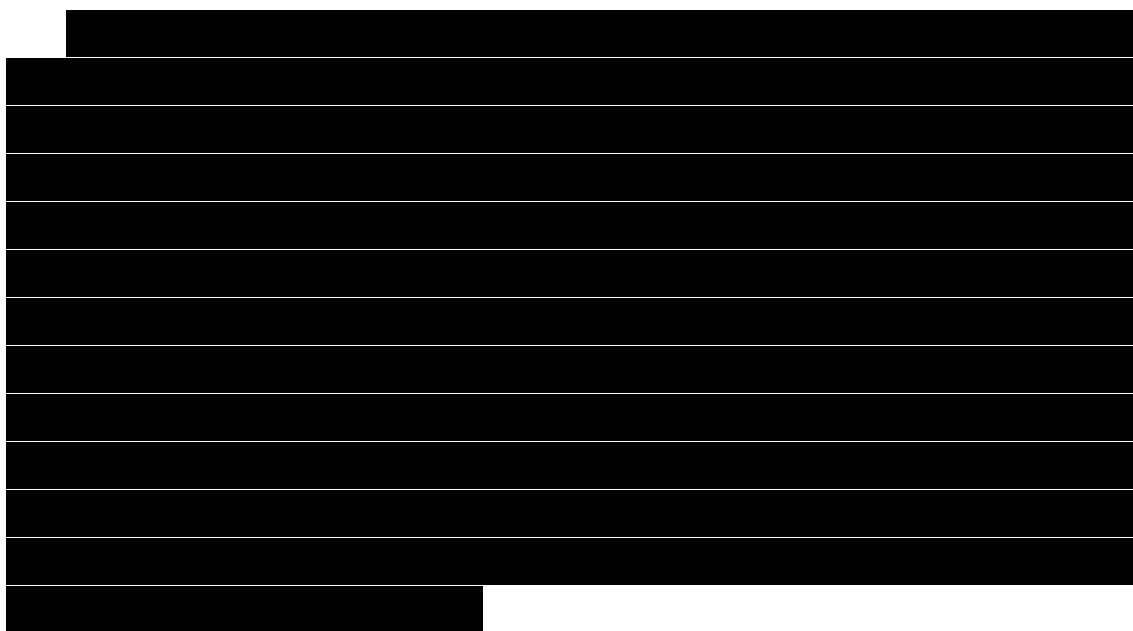


Figura. 58 Ecrã de “Detalhe de transfusão”

Capítulo 5 Considerações finais

Neste capítulo são apresentadas as conclusões retiradas deste PEI, as dificuldades encontradas e o trabalho futuro.

5.1 Trabalho realizado

Os sistemas de informação na área da saúde são atualmente ferramentas imprescindíveis na prestação de cuidados de saúde. Potenciam não só a tomada de melhores decisões médicas, na medida em que permite uma maior disponibilidade de informações clínicas num dado momento, como também o aumento da segurança e bem-estar dos pacientes, e a melhoria da saúde pública, recorrendo à investigação e monitorização dos dados de saúde. Foi neste contexto e necessidade que surgiu a aplicação *Clinidata® Transfusion*, tema deste PEI.

O trabalho realizado no contexto deste PEI teve como objetivo o desenvolvimento do módulo *Clinidata® Transfusion* para ser integrado no *Clinidata®*. Este módulo gere automaticamente dadores, transfusões e bancos de sangue para ser utilizado pelo médico, enfermeiro e/ou técnico de laboratório. Neste contexto, fui responsável pela implementação do módulo de gestão de dadores, as suas tabelas, ecrãs, modelos de domínios, *repository* e *service* e dos ecrãs de gestão e detalhe de transfusões. Também fui responsável por correções detetadas em fase de testes pela equipa de assistência no módulo de gestão de transfusões ou alterações que tenham sido pedidas pelo cliente. O desenvolvimento deste módulo seguiu o documento de análise do banco de sangue, que foi realizado a partir do levantamento de requisitos dos clientes pelos analistas seniores. Tendo em conta a lista de requisitos funcionais (secção 3.1.1) para este módulo de dadores, penso que o trabalho foi bem-sucedido e que objetivo foi atingido uma vez que foram cumpridos todos os requisitos inerentes às funcionalidades que implementei.

Com a realização deste PEI foi possível adquirir experiência profissional, integrar numa equipa dinâmica de desenvolvimento de um sistema de gestão de informação da saúde, por em prática competências adquiridas ao longo deste percurso académico, adquirir novas competências e contribuir para um projeto desafiante e ambicioso, com o propósito técnico de melhorar o desempenho dos seus utilizadores nas suas função diárias, que por sua vez tem impacto direto nos pacientes.

Para além disso, tive a oportunidade de colocar em prática vários conceitos adquiridos no decurso académico -uso do *GIT*, *Spring*, *Hibernate*, modelo MVP e uso de HQL. O facto de ter implementado algo de raiz nas três camadas da aplicação (camada cliente, camada negócio, camada de dados) permitiu-me adquirir um maior

conhecimento sobre a lógica de negócio, nomenclaturas e metodologias usadas na Maxdata. Com este projeto tive a oportunidade de conhecer e usar a ferramenta GWT, que nos permite desenvolver uma aplicação web em *front-end* e *back-end* em *Java*, tirando da responsabilidade do programador as particularidades do *browser* e da linguagem de *JavaScript*.

5.2 Dificuldades encontradas

Durante o desenvolvimento deste PEI, houve bastantes alterações ao documento de análise do banco de sangue quando já estava a implementar o módulo de dadores, levando a que tivesse que reprogramar várias vezes o mesmo requisito. Foi necessário fazer e refazer várias tabelas, relações de tabelas, aspetos visuais do ecrã ou determinados comportamentos que já estavam implementados.

5.3 Trabalho futuro

O trabalho implementado no contexto deste PEI vai começar a ser testado pela equipa de assistência, podendo levar a que seja necessário proceder a correções. Se não forem detetados erros, então vai ser apresentada aos clientes. Nesta apresentação podem surgir mudanças nos requisitos e como tal pode ser necessário alterar ou pedidos de novas funcionalidades ou comportamento do software por parte dos clientes. Como tal o trabalho futuro passa pela manutenção ou implementação de novas funcionalidades ao módulo de dadores.

Capítulo 6 Bibliografia

- [1] “Número de doadores em Portugal” [Online]. Disponível em: <https://www.dn.pt/vida-e-futuro/interior/209-mil-portugueses-dao-sangue-jovens-sao- apenas-14-10721730.html>
- [2] M. Alexandre, S. Ana, T. Rosa, “Manual de triagem de doadores de sangue”, 2014, pp12-87.
- [3] F. Sousa, L. Moreira, J. Machado, “Computação em nuvem: conceitos, tecnologias aplicações e desafios”, cap.7.
- [4] L. Wei, P. Eun, “e-healthcare cloud-enabling characteristics, challenges and adaptation solutions”, 2013.
- [5] D. Troung, “How Cloud Computing Enhances Competitive advantages: A research model for small business”, 2015.
- [6] Serviço Nacional de Saúde. Especificações do Sistema de Qualidade dos Serviços de Sangue e Serviços de Medicina Transfusional [Online]. Disponível em: http://www.ipst.pt/files/IPST/INFORMACAO_DOCUMENTACAO/Norma_21_2017_DGS_especificacoes_sistema_qualidade_servicos_sangue_medicina_transfusional.pdf
- [7] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.
- [8] D. Lewis, “What is the web 2.0?” Crossroads, vol. 13, no. 1, p. 3, 2006.
- [9] “Introdução ao ISBT-128” [Online]. Disponível em: <http://www.iccbba.org/uploads/77/b5/77b5b71bc50f4634607797b03b22d78b/IN-028-An-Introduction-to-ISBT-128-Portuguese-v1.pdf>
- [10] “*Clinidata@BST* Gestão automática de banco de sangue e transfusões” [Online]. Disponível em: http://maxdata.pt/site/uploads/noticias/documentos/20121018181040-1maxdata_newsletter_5.pdf
- [11] P. Maria, “Cuidar com segurança na administração de componentes sanguíneos”, 2013
- [12] Ministério da Saúde. Tipos sanguíneos – Sistema AB0 e Fator RH [Online]. Disponível em: <https://www.mdsaude.com/2017/01/tipos-sanguineos-sistema-abo.html>
- [13] “Serviço nacional de saúde” [Online]. Disponível em: <http://ipst.pt>

- [14] Site oficial de dador Português [Online]. Disponível em: <http://www.dador.pt>
- [15] S. Gaurav, P. Anil, R. Jay, T. Darrell, “Contemporary issues in transfusion medicine informatics”
- [16] Roger S. Pressman, Software engineering: a practitioner’s approach, 5th ed. 2001.
- [17] A. Bjork, “Agile Development and the Daily Stand-up Meeting,” Visual Studio Magazine, 2011.
- [18] “Code Reviews, what are the advantages?” [Online]. Disponível em: <http://programmers.stackexchange.com/questions/71654/code-reviews-what-are-the-advantages>.
- [19] J. Atwood, “Code Reviews: Just Do It,” 2006. [Online]. Disponível em: <http://blog.codinghorror.com/code-reviews-just-do-it/>
- [20] Sun quest Laboratory Blood Bank Module. [Online]. Disponível em: <https://www.sunquestinfo.com/laboratory-pathology-informatics/blood-bank-management/>
- [21] “TDBlood Bank Transfusion the right product to the tight patient Newspapers”
- [22] HEMOVIDA - Sistema de Gerenciamento em Serviços de Hemoterapia. [Online]. Disponível em: <http://datasus.saude.gov.br/sistemas-e-aplicativos/hospitalares/hemovida>
- [23]”SBS.Web Sistema de banco de sangue” [Online]. Disponível em: <http://www.sbs.srv.br/index.php/component/content/article/77-sbs/69-sistemabancosangue>
- [23] ASIS base de dados do sangue” [Online]. Disponível em: <http://www.rcc.gov.pt/Directorio/Temas/IG/Paginas/Base-de-Dados-do-Sangue-.aspx>
- [24] J. A. Jacko and A. Sears, Eds., The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 2003.
- [25] J. A. Jacko and A. Sears, Eds., The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 2003
- [26] “GWT overview [Online]. Disponível em: <http://www.gwtproject.org/overview.html>.
-

[27] R. Ramakrishnan and J. Gehrke, Database Management Systems, 2nd Ed. New York, NY, USA: McGraw-Hill, Inc., 1999.

[28] J. Shannon, B. Adida, and D. Baccus, “Oracle to Postgres Conversion.” [Online]. Disponível em:
http://wiki.postgresql.org/wiki/Oracle_to_Postgres_Conversion#Grammar_Differences.

[29] P. Avgeriou and U. Zdun, “Architectural Patterns Revisited—A Pattern Language,” in the 10th European Conference on Pattern Languages of Programs (EuroPlop 2005), 2005, pp. 1–39.

[30] S. Chandel, “Testing Methodologies Using Google Web Toolkit,” 2009. [Online]. Disponível em:
https://developers.google.com/webtoolkit/articles/testing_methodologies_using_gwt

Capítulo 7 Anexos

7.1 Diagrama de relações de dados

